RÉPUBLIQUE ALGÉRIENNE DÉMOCRATIQUE ET POPULAIRE
MINISTÈRE DE L'ENSEIGNEMENT SUPÉRIEUR ET DE LA RECHERCHE SCIENTIFIQUE
UNIVERSITÉ ABOU BEKR BELKAID TLEMCEN
FACULTÉ DE TECHNOLOGIE
DÉPARTEMENT DE TELECOMMUNICATIONS

THÈSE
Présentée pour l'obtention du diplôme de
DOCTORAT
en TELECOMMUNICATIONS
Spécialité : Communication et réseaux sans fil

Par
M^r **BOUIDAINE Al Baraa**

Détection intelligente des intrusions système basée sur le Deep Learning

---

Thèse soutenue publiquement en 2025 devant le jury composé de :

Pr. Merzougui Rachid	Université de Tlemcen	Président
Dr. Merzoug Mohammed	Université de Tlemcen	Examinateur
Dr. Souiki Sihem	Université Ain-Temouchent	Examinateur
Pr. Hadjila Mourad	Université de Tlemcen	Directeur de thèse
Dr. Moussaoui Djilali	Université de Tlemcen	Invité

Année universitaire 2025-2026

Remerciements

Louange à Allah qui ma donné patience et courage pour mener à bien ce travail de thèse malgré les difficultés rencontrées. Je tiens à remercier Mr Hadjila Mourad, mon directeur de thèse pour leurs précieux conseils.

Je remercie également les membres du jury qui ont accepté l'évaluation de notre travail, et je cite en l'occurrence Pr Merzougui Rachid, Dr Merzoug Mohammed, Dr Souiki Sihem, et Dr Moussaoui Djilali.

Je remercie aussi ma famille pour son soutien, son écoute et ses encouragements tout au long de cette thèse.

Je dédie ce modeste travail
A mes très chers parents, qu'Allah tout puissant les protège
A ma très chère femme, mon enfant, mon frère et mes soeurs
A mes collègues

Résumé

Cette thèse traite de la sécurité des réseaux informatiques et de l'application du Deep Learning à la détection d'intrusions. Elle s'ouvre sur une présentation des fondements de la cybersécurité, des principales menaces comme les attaques DoS, DDoS, brute force, injections SQL et botnets, ainsi que des mécanismes de protection tels que les pare-feux, le cryptage, les VPN et les systèmes de détection d'intrusions (IDS). Elle introduit ensuite les concepts d'intelligence artificielle, de machine learning et de deep learning, en décrivant les architectures neuronales (ANN, CNN, RNN) et les métriques d'évaluation utilisées pour l'analyse et la classification des attaques. L'étude expérimentale porte sur la détection d'attaques DDoS à l'aide de modèles de deep learning appliqués aux datasets CSE-CIC-IDS2018 et Edge-IIoTSet, avec une comparaison des performances selon différents types de classification. Enfin, une approche améliorée de détection d'anomalies IDS est proposée, intégrant des techniques d'optimisation et de réduction de dimension afin d'accroître la précision et la robustesse du modèle. Les résultats démontrent l'efficacité du deep learning dans la détection automatisée des intrusions et ouvrent la voie à des modèles plus légers et adaptatifs pour les environnements IoT et Edge Computing.

Mots clés : Sécurité réseaux, Deep Learning, DDoS, Détection d'intrusion, Intelligence Artificielle, détection d'attaques, classification, datasets CSE-CIC-IDS20218 et Edge-IIoTSet.

Abstract

This thesis addresses the security of computer networks and the application of Deep Learning to intrusion detection. It begins with a presentation of the foundations of cybersecurity, the main threats such as DoS, DDoS, brute force attacks, SQL injections, and botnets, as well as protection mechanisms including firewalls, encryption, VPNs, and intrusion detection systems (IDS). It then introduces the concepts of artificial intelligence, machine learning, and deep learning, describing neural architectures (ANN, CNN, RNN) and evaluation metrics used for the analysis and classification of attacks. The experimental study focuses on detecting DDoS attacks using deep learning models applied to the CSE-CIC-IDS2018 and Edge-IIoTSet datasets, with a performance comparison across different types of classification. Finally, an enhanced IDS anomaly detection approach is proposed, integrating optimization and dimensionality reduction techniques to improve the models accuracy and robustness. The results demonstrate the effectiveness of deep learning in automated intrusion detection and pave the way for lighter and more adaptive models suitable for IoT and Edge Computing environments.

Keywords: Network security, deep learning, DDoS, intrusion detection, artificial intelligence, attack detection, classification, CSE-CIC-IDS20218 and Edge-IIoTSet datasets.

ملخص

تتناول هذه الأطروحة أمن شبكات الحاسوب وتطبيقات التعلم العميق في كشف التسلل. تبدأ باستعراض عام لأساسيات الأمن السيبراني، بما في ذلك التهديدات الرئيسية مثل هجمات الحرمان من الخدمة (DoS)، وهجمات الحرمان من الخدمة الموزعة (DDoS)، وهجمات القوة الغاشمة، وحقن SQL، وشبكات الروبوتات، بالإضافة إلى آليات الحماية مثل جدران الحماية، والتشفير، وشبكات VPN، وأنظمة كشف التسلل (IDS). ثم تُقدم مفاهيم الذكاء الاصطناعي، والتعلم الآلي، والتعلم العميق، وتصف بنى الشبكات العصبية (ANNs، وCNNs، وRNNs) ومقاييس التقييم المستخدمة في تحليل الهجمات وتصنيفها. تُركز الدراسة التجريبية على كشف هجمات الحرمان من الخدمة الموزعة (DDoS) باستخدام نماذج التعلم العميق المُطبقة على مجموعات بيانات CSE-CIC-IDS2018 وEdge-IIoTSet، مع مقارنة الأداء عبر طرق تصنيف مختلفة. وأخيرًا، يُقترح نهج مُحسّن للكشف عن شذوذ أنظمة كشف التسلل، يتضمن تقنيات التحسين وتقليل الأبعاد لزيادة دقة النموذج ومتانته. تُظهر النتائج فعالية التعلم العميق في الكشف الآلي عن التسلل، وتمهد الطريق لنماذج أخف وزنًا وأكثر تكيفًا لبيئات إنترنت الأشياء والحوسبة الطرفية.

الكلمات المفتاحية: أمن الشبكات، التعلم العميق، هجمات الحرمان من الخدمة الموزعة (DDoS)، كشف التسلل، الذكاء الاصطناعي، كشف الهجمات، التصنيف، مجموعات بيانات CSE-CIC-IDS20218 وEdge-IIoTSet.

Table des matières

Table des figures

Liste des tableaux

Introduction générale

L'évolution rapide vers un mode de vie entièrement connecté pour l'ensemble des acteurs économiques et sociaux a engendré des avantages considérables en matière de communication, de commerce en ligne et d'échange d'informations. Cette interconnexion généralisée favorise l'accès instantané aux services numériques, facilite les transactions commerciales globalisées et optimise le partage de connaissances à l'échelle mondiale. Toutefois, cette dépendance accrue aux technologies de l'information et de la communication s'accompagne de vulnérabilités critiques et de défis majeurs en matière de cybersécurité. En effet, les menaces cybernétiques évoluent à un rythme soutenu, les attaquants exploitant en permanence les failles et les points faibles des systèmes informatiques pour mener diverses formes d'attaques, telles que les intrusions non autorisées, le vol de données sensibles, le sabotage des infrastructures numériques ou encore la compromission des réseaux d'entreprises. Cette dynamique complexe impose aux organisations et aux individus de renforcer en permanence leurs dispositifs de sécurité afin de faire face à un paysage de menaces de plus en plus sophistiqué et imprévisible.

L'ampleur du problème est particulièrement préoccupante, avec environ 2328 cybercrimes recensés chaque jour et des pertes estimées à près de 26 milliards de dollars au cours des 21 dernières années [1]. Face à cette recrudescence des menaces numériques et à la sophistication croissante des techniques d'attaque, les experts en cybersécurité ont développé des systèmes de détection d'intrusion (Intrusion Detection Systems, IDS) afin de surveiller en continu le trafic réseau à la recherche de tout comportement anormal ou usage abusif des ressources informatiques. Les IDS se sont imposés comme des outils indispensables dans la lutte contre les cyberattaques en offrant une capacité de détection précoce des menaces potentielles avant qu'elles ne puissent compromettre l'intégrité, la confidentialité ou la disponibilité des systèmes ciblés. En identifiant rapidement les activités suspectes, ces systèmes permettent aux administrateurs de réseau de réagir de manière proactive et de mettre en uvre des mesures correctives appropriées, contribuant ainsi à renforcer la résilience des infrastructures numériques face aux attaques toujours plus complexes des cybercriminels.

Les systèmes de détection d'intrusion (Intrusion Detection Systems, IDS) peuvent

être globalement classés en deux grandes catégories : les IDS basés sur le réseau (Network-Based Intrusion Detection Systems, NIDS) et les IDS basés sur l'hôte (Host-Based Intrusion Detection Systems, HIDS). Les NIDS assurent la surveillance du trafic réseau afin de détecter toute tentative de compromission visant à dévier le fonctionnement normal du système informatique. Ils analysent en temps réel les paquets de données circulant sur le réseau à la recherche de signatures d'attaques connues, d'anomalies comportementales ou de schémas suspects indiquant une activité malveillante. En revanche, les HIDS se concentrent sur l'analyse des événements se produisant localement sur la machine où ils sont installés. Ils examinent notamment les journaux système, les fichiers critiques, les processus en exécution, et les modifications de configuration susceptibles de révéler une intrusion ou une utilisation abusive des ressources locales. Cette complémentarité entre les NIDS et les HIDS permet de couvrir à la fois les attaques externes transitant par le réseau et les compromissions internes survenant directement au sein des systèmes hôtes.

Au cours des dernières années, l'apprentissage profond (Deep Learning, DL) s'est imposé comme un outil particulièrement performant dans le domaine de la détection d'intrusion. Les modèles de DL, tels que les réseaux neuronaux convolutifs (Convolutional Neural Networks, CNN) et les réseaux neuronaux récurrents (Recurrent Neural Networks, RNN), sont capables d'apprendre automatiquement des schémas complexes à partir de volumes massifs de données, et d'identifier des menaces potentielles sans recourir à des règles explicites ou à des signatures prédéfinies. Les systèmes de détection d'intrusion basés sur l'apprentissage profond ont ainsi démontré des résultats prometteurs pour la détection de divers types d'attaques réseau, incluant les attaques par déni de service (Denial of Service, DoS), les infections par logiciels malveillants (malware), ainsi que les intrusions réseau sophistiquées. Grâce à leur capacité d'apprentissage à partir des historiques de trafic réseau, ces modèles sont en mesure de reconnaître des attaques inédites ou en constante évolution. Cette aptitude à détecter des menaces inconnues revêt une importance particulière dans le contexte des menaces émergentes, où les méthodes traditionnelles fondées sur des signatures peinent souvent à identifier de nouveaux vecteurs d'attaque ou des variantes encore non répertoriées.

En définitive, bien que les avantages d'un mode de vie entièrement connecté soient indéniables, il demeure essentiel de prendre pleinement conscience des risques engendrés par les menaces cybernétiques. Face à l'évolution rapide et à la sophistication croissante des attaques, le recours aux systèmes de détection d'intrusion (IDS) ainsi qu'aux modèles basés sur l'apprentissage profond (DL) offre des perspectives prometteuses pour renforcer la sécurité des réseaux et des systèmes d'information. Ces approches permettent de détecter et de neutraliser plus efficacement les menaces émergentes, contribuant ainsi à une meilleure protection des infrastructures numériques et des utilisateurs dans un environnement technologique en perpétuelle mutation.

1. **Motivation**

   La croissance rapide des réseaux informatiques modernes, l'explosion des volumes de données échangées et la sophistication croissante des attaques informatiques ont considérablement complexifié la tâche de sécurisation des infrastructures numériques. Les systèmes de détection d'intrusion (IDS), qui constituent l'une des premières lignes de défense face aux cyberattaques, sont confrontés à des limitations majeures lorsqu'ils reposent sur des approches traditionnelles. Les méthodes classiques fondées sur les signatures nécessitent des bases de connaissances constamment mises à jour et s'avèrent souvent inefficaces face aux attaques nouvelles ou inconnues (zero-day attacks). Les approches basées sur des règles sont quant à elles fortement dépendantes de l'expertise humaine et présentent un risque élevé de faux positifs dans des environnements complexes et dynamiques.

   Dans ce contexte, l'intégration des techniques d'apprentissage profond (Deep Learning) dans les IDS apparaît comme une solution particulièrement prometteuse. Le Deep Learning permet aux modèles de s'affranchir de l'extraction manuelle des caractéristiques en apprenant automatiquement des représentations complexes et hiérarchiques directement à partir des données brutes de trafic réseau. Grâce à leur capacité à modéliser des relations non linéaires complexes et à capturer des patterns comportementaux subtils, les modèles de Deep Learning offrent des performances accrues pour la détection des attaques connues et surtout pour l'identification des comportements anormaux et des menaces émergentes.

   De plus, les architectures avancées du Deep Learning, telles que les réseaux neuronaux convolutifs (CNN), les réseaux récurrents (RNN, LSTM, GRU) et leurs combinaisons hybrides, sont capables de traiter efficacement à la fois les aspects spatiaux et temporels des flux réseau, ce qui est crucial pour détecter des attaques distribuées ou progressives. Par ailleurs, le Deep Learning contribue à la réduction des taux de faux positifs, un problème récurrent dans les IDS traditionnels, en améliorant la précision et la robustesse des systèmes de détection.

   Enfin, dans un contexte où les volumes de données sont massifs et en constante augmentation, le Deep Learning constitue une approche scalable, apte à traiter des flux continus de données en temps réel tout en conservant des performances de détection élevées. Cette capacité à s'adapter à l'évolution rapide des menaces et à l'hétérogénéité croissante des environnements numériques justifie pleinement l'intérêt croissant accordé à l'intégration du Deep Learning dans les systèmes de détection d'intrusion de nouvelle génération.

2. **Organisation de la thèse**

   Cette thèse est organisée en quatre chapitres comme suit :

- Dans le premier chapitre introductif intitulé " Sécurité des réseaux et des systèmes de détection d'intrusion ", un état de l'art détaillé de la sécurité des réseaux est présenté, incluant les principes fondamentaux tels que la confidentialité, l'intégrité et la disponibilité. Une classification exhaustive des principales menaces est exposée : attaques par déni de service (DoS/DDoS), attaques par force brute, injections SQL, infiltrations, et botnets. Ensuite, les mécanismes de protection courants sont détaillés, allant des pare-feux aux VPN, en passant par le chiffrement, le contrôle d'accès et les systèmes SIEM. Une attention particulière est accordée aux systèmes de détection d'intrusion (IDS), avec une distinction entre les IDS basés sur le réseau (NIDS), sur l'hôte (HIDS) et hybrides. Les différentes méthodes de détection (signature, anomalies) ainsi que les architectures fonctionnelles des IDS sont également expliquées.
- Le deuxième chapitre intitulé " Intelligence artificielle, Machine Learning et Deep Learning " expose les fondements théoriques de l'intelligence artificielle appliquée à la cybersécurité. Après avoir introduit les principes de l'apprentissage automatique supervisé (modèle, dataset, fonction de coût, algorithmes), l'accent est mis sur l'apport du Deep Learning dans les systèmes IDS. Les architectures des réseaux neuronaux sont étudiées, incluant les réseaux neuronaux feedforward (FFNN), les réseaux convolutifs (CNN) pour l'extraction spatiale des caractéristiques, et les réseaux récurrents (RNN, LSTM, GRU) pour la modélisation temporelle. Les avantages du DL en termes de généralisation, d'automatisation de l'extraction des caractéristiques et de détection des attaques inconnues sont discutés.
- Dans le troisième chapitre expérimental intitulé " Détection des attaques DDoS basée sur le Deep Learning via les datasets CSE-CIC-IDS2018 et Edge-IIoTset ", deux bases de données de référence sont exploitées. Après une présentation des jeux de données CSE-CIC-IDS2018 et Edge-IIoTset, un important travail de prétraitement est détaillé : nettoyage des données, codage des variables catégorielles, normalisation et sélection des caractéristiques optimales. Différentes approches de classification sont explorées (binaire, multi-classes, multi-label). Des modèles de Deep Learning sont entraînés sur ces données, et leurs performances sont évaluées à travers des métriques standards : exactitude, précision, rappel et score F1.
- Le quatrième chapitre intitulé "Détection d'Anomalies pour les Systèmes de Prévention d'Intrusion basée sur l'Apprentissage Profond (CSE-CIC-IDS2018)", s'organise selon une progression logique commençant par une introduction présentant le sujet et ses enjeux. Il se poursuit par une revue des travaux connexes, structurée autour des taxonomies des méthodes IDS, des applications des réseaux neuronaux profonds, des techniques de

détection ciblée et d'optimisation, des méthodes de réduction de dimension et des approches spécialisées à hautes performances. La méthodologie adoptée est ensuite détaillée, décrivant le jeu de données utilisé, les étapes de prétraitement (fusion, nettoyage, codage, normalisation et division des données), les métriques d'évaluation et la création du modèle. Le chapitre présente ensuite les résultats et leurs analyses, incluant une approche améliorée, avant de se conclure par une synthèse des apports et des perspectives.

Ce manuscrit est clôturé par une conclusion générale qui rappelle les différentes contributions élaborées dans le cadre de ce travail de recherche et présente quelques perspectives pour les futures travaux de recherche.

Chapitre 1

Sécurité des réseaux et système de détection des intrusions

Sommaire

1.1 Introduction

La sécurité réseau constitue un élément fondamental de l'architecture de cybersécurité, regroupant des mécanismes, politiques et protocoles destinés à assurer la confidentialité, l'intégrité et la disponibilité des données (CIA triad) circulant dans un système informatique distribué. Ces trois piliers sont reconnus comme les principes directeurs de toute stratégie de sécurité informatique [2].

Dans ce cadre, les Systèmes de Détection d'Intrusion (IDS) jouent un rôle crucial. Contrairement aux pare-feux, qui opèrent en mode préventif en bloquant les connexions non autorisées, les IDS adoptent une approche passive et réactive, en inspectant le trafic réseau à la recherche d'anomalies ou de comportements suspects

susceptibles d'indiquer une attaque en cours. Ils alertent les administrateurs en cas de détection d'activités malveillantes, facilitant ainsi une réponse rapide [3].

Deux grandes catégories d'IDS coexistent : les systèmes à base de signatures, capables de reconnaître des attaques connues, et ceux à base comportementale ou d'apprentissage automatique, capables d'identifier des menaces inconnues, notamment les attaques zero-day ou les mouvements latéraux dans un réseau compromis [4, 5]. Ces approches hybrides renforcent leur efficacité dans un contexte de cybermenaces sophistiquées, persistantes et évolutives.

L'intégration des IDS dans l'écosystème de cybersécurité est aujourd'hui indispensable pour garantir une surveillance en continu, réduire la surface d'attaque, et améliorer la résilience des infrastructures critiques face aux menaces persistantes avancées (APT) et autres vecteurs d'intrusion [6].

1.2 Les piliers de la sécurité des réseaux

La triade CIA, qui signifie Confidentialité, Intégrité et Disponibilité, est un concept fondamental de la sécurité des réseaux [7, 8]. Il s'agit d'un ensemble de principes directeurs et d'objectifs permettant aux organisations et aux individus de protéger les informations contre les accès non autorisés, les modifications ou les pertes. Pour assurer une véritable sécurité des réseaux, les trois éléments de la triade CIA doivent être présents simultanément. Par conséquent, la triade CIA est considérée comme une base essentielle pour des pratiques efficaces en matière de sécurité de l'information (voir figure 1.1).

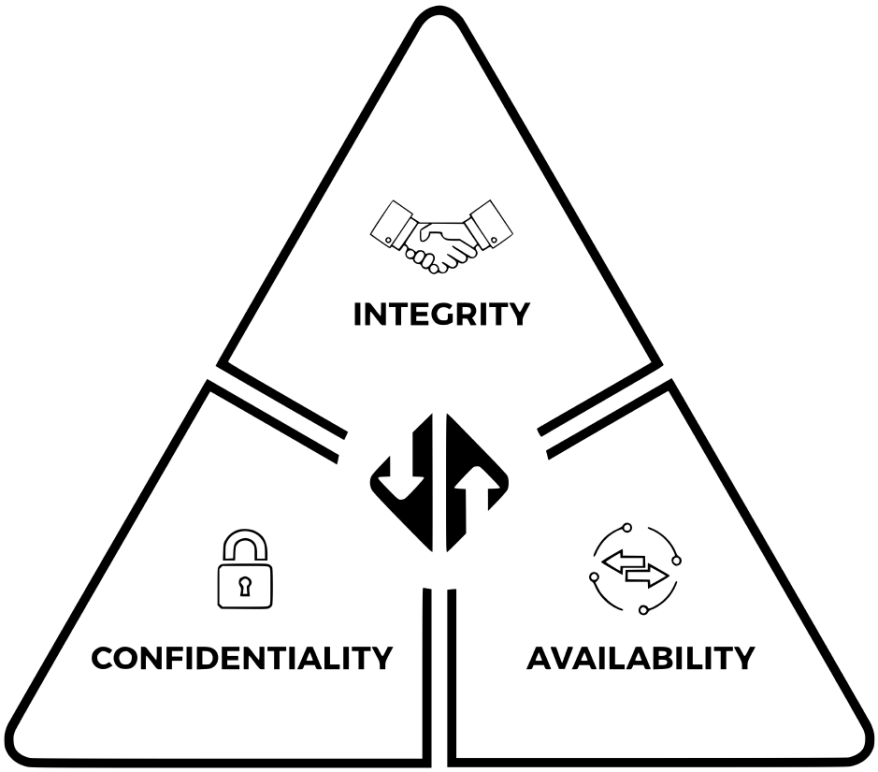


Figure 1.1 *– Triade CIA.*

1.2.1 Confidentialité

La confidentialité constitue l'un des trois piliers fondamentaux de la triade CIA (Confidentialité, Intégrité, Disponibilité), et représente l'obligation de restreindre l'accès à l'information aux seules personnes ou entités autorisées. Elle vise à empêcher toute divulgation, consultation ou interception non autorisée de données sensibles, que ce soit dans un cadre organisationnel, institutionnel ou individuel. La mise en uvre de la confidentialité est d'autant plus critique que le niveau de sensibilité des informations manipulées est élevé, notamment dans les domaines de la santé, de la défense, de la finance ou encore des télécommunications.

Afin de garantir cette confidentialité, divers mécanismes de sécurité techniques et organisationnels sont généralement mis en uvre. Parmi les plus couramment utilisés, on retrouve :

1. Le chiffrement (cryptographie), qui consiste à transformer des données en un format illisible sans la possession d'une clé de déchiffrement appropriée. Le chiffrement symétrique (comme AES [9]) ou asymétrique (comme RSA [10]) permet de sécuriser les échanges de données, aussi bien en transit que stockées.
2. L'authentification à plusieurs facteurs (MFA Multi-Factor Authentication [11]), combinant généralement un mot de passe (facteur de connaissance), un appareil personnel (facteur de possession), et une caractéristique biométrique (facteur

d'inhérence), renforce le niveau de sécurité en limitant les risques liés au vol d'identifiants.

3. Les mots de passe robustes et la gestion des identifiants, essentiels pour restreindre l'accès aux seuls utilisateurs autorisés. Leur complexité, renouvellement périodique et stockage sécurisé participent à la protection des ressources.
4. L'identification biométrique, basée sur des caractéristiques physiques ou comportementales uniques (empreinte digitale, reconnaissance faciale, scan rétinien), constitue un moyen d'authentification avancé de plus en plus répandu, notamment dans les systèmes critiques.
5. Les jetons de sécurité (security tokens), physiques ou logiciels, permettent d'ajouter une couche supplémentaire de vérification lors des processus d'accès à des systèmes sensibles.
6. Les politiques d'accès basées sur les rôles (RBAC Role-Based Access Control [12]), qui permettent d'attribuer des droits d'accès strictement nécessaires à chaque utilisateur en fonction de sa fonction ou responsabilité dans l'organisation.

Ces différentes mesures, lorsqu'elles sont bien intégrées dans une politique de sécurité de l'information, permettent de minimiser les risques de fuites de données, d'espionnage industriel, ou d'atteintes à la vie privée. Il est crucial de calibrer le niveau de protection en fonction de la classification de la sensibilité des informations, comme cela est recommandé dans les cadres normatifs tels que le NIST SP 800-53 [13] ou l'ISO/IEC 27001 [14].

En somme, la préservation de la confidentialité repose sur une combinaison de techniques cryptographiques, de contrôles d'accès rigoureux et de politiques de sécurité adaptées à la criticité des données concernées. Une faille dans la confidentialité peut avoir des conséquences graves, allant de la compromission de secrets commerciaux à des sanctions réglementaires sévères en cas de violation de données personnelles, notamment sous des cadres légaux comme le RGPD ou la HIPAA [15].

1.2.2 Intégrité

L'intégrité désigne la préservation de l'exactitude, de la cohérence et de l'exhaustivité des données tout au long de leur cycle de vie, garantissant ainsi que les informations ne soient modifiées que par des actions autorisées et intentionnelles. Elle implique la protection contre les modifications non autorisées, les altérations accidentelles ainsi que les perturbations d'origine non humaine telles que les défaillances matérielles ou les pannes de serveurs, susceptibles de compromettre la fiabilité des données. Pour assurer l'intégrité des données, un ensemble de mesures techniques et administratives peut être mis en uvre. Parmi celles-ci figurent les techniques cryptographiques telles que le chiffrement et le hachage pour sécuriser le contenu des

données, les contrôles d'accès des utilisateurs pour limiter les droits de modification, l'utilisation de sommes de contrôle (checksums) et de systèmes de gestion des versions pour surveiller et vérifier les modifications, ainsi que la réalisation de sauvegardes régulières permettant de restaurer les données originales en cas de corruption ou de perte. Le choix et le déploiement de ces mesures doivent être soigneusement adaptés à la sensibilité, à la criticité et au contexte opérationnel des données, afin de garantir leur fiabilité et leur utilité, tant dans les opérations courantes que dans les processus décisionnels critiques [10].

1.2.3 Disponibilité

La disponibilité vise à garantir que les utilisateurs autorisés puissent accéder aux données et aux services informatiques chaque fois qu'ils en ont besoin, sans interruption ni retard. Elle implique la mise en uvre de multiples mesures préventives et correctives destinées à assurer le fonctionnement continu des systèmes, même en cas de défaillance ou de surcharge. Parmi ces mesures figurent la redondance des ressources matérielles et logicielles afin de pallier les défaillances potentielles, l'équilibrage de charge (load balancing) pour répartir de manière optimale les requêtes entre différents serveurs, la planification de la reprise après sinistre permettant de restaurer rapidement les services en cas de catastrophe, ainsi que l'entretien régulier et la surveillance constante des infrastructures pour détecter et corriger proactivement les anomalies avant qu'elles ne provoquent des pannes. En garantissant la disponibilité des services réseau, les organisations et les individus peuvent éviter les pertes de productivité, les pertes financières et d'autres conséquences néfastes qui résultent des interruptions de service ou des périodes d'indisponibilité [8].

En plus de la triade CIA, il existe un autre ensemble de mesures qui doivent être mises en place afin de garantir la sécurité de l'information. Ces mesures sont connues sous les termes d'authentification, d'autorisation et de traçabilité (ou comptabilité des accès).

1.3 Menaces pour la sécurité des réseaux

Les réseaux informatiques sont exposés à une multitude de menaces de sécurité susceptibles de compromettre la confidentialité, l'intégrité et la disponibilité des données. Parmi ces menaces figurent notamment les virus, les logiciels malveillants (malwares), ainsi que les attaques par piratage informatique (hacking). Ces attaques peuvent entraîner des conséquences graves telles que la perte ou la corruption des données, des interruptions prolongées des services informatiques, des dégradations de performances du système, ainsi que des pertes financières significatives pour les organisations concernées. De plus, les atteintes à la sécurité des réseaux peuvent également porter préjudice à la réputation des entreprises et engendrer des risques juridiques.

Il est donc essentiel de bien connaître les différentes formes d'attaques auxquelles un réseau peut être confronté afin de mettre en place des mesures de protection adaptées [17]. Parmi les attaques les plus courantes figurent notamment celles que nous présentons ci-après.

1.3.1 Déni de service (DoS)

L'attaque par déni de service (DoS) constitue une forme d'attaque informatique qui vise à perturber le fonctionnement normal d'un ordinateur, d'un serveur ou de tout autre dispositif en le submergeant de trafic, le rendant ainsi indisponible pour ses utilisateurs légitimes [18]. Ce type d'attaque peut être initié à partir d'une seule machine et repose généralement sur l'envoi massif de requêtes vers la cible, jusqu'à saturer ses capacités de traitement, empêchant ainsi le traitement du trafic légitime. L'objectif principal d'une attaque DoS est donc d'épuiser les ressources du système visé, telles que la mémoire, la bande passante ou la capacité de calcul, provoquant ainsi une interruption de service pour les utilisateurs autorisés. De telles attaques peuvent engendrer des perturbations considérables pour les entreprises et les organisations, affectant leur productivité, leurs revenus et leur réputation. Afin de prévenir et d'atténuer les effets de ces attaques, diverses mesures de protection peuvent être mises en uvre, notamment des dispositifs de filtrage de trafic, des systèmes de détection d'intrusion, des mécanismes de répartition de charge et des stratégies de limitation de débit. La Figure 1.2 illustre le fonctionnement d'une attaque DoS.

Figure 1.2 – *Attaque DoS.*

1.3.1.1 Hulk

HULK (HTTP Unbearable Load King) est un outil d'attaque de type déni de service (DoS) qui fonctionne en envoyant un nombre extrêmement élevé de requêtes HTTP vers un serveur web dans le but de le saturer et de le rendre incapable de répondre aux demandes légitimes de ses utilisateurs [19]. Le principe de cette attaque repose sur la génération dynamique de requêtes HTTP variées afin de contourner les mécanismes de cache et de détection des attaques basés sur des signatures statiques. Cette capacité à produire des requêtes toujours différentes rend l'attaque particulièrement difficile à détecter et à bloquer à l'aide des méthodes de filtrage traditionnelles. Le nom ń HULK ż fait référence au personnage de bande dessinée de Marvel Comics,

soulignant ainsi l'aspect brutal et massif de l'attaque, conçue pour ń fracasser ż les défenses des serveurs web par une surcharge intentionnelle de trafic. De par sa simplicité d'utilisation et son efficacité redoutable, HULK représente une menace sérieuse pour les infrastructures web mal protégées.

1.3.1.2 GoldenEye

GoldenEye est un autre outil d'attaque de type déni de service (DoS) qui, à l'instar de HULK, envoie un grand nombre de requêtes HTTP vers un serveur web dans le but de le saturer et de le rendre indisponible pour les utilisateurs légitimes [20]. Toutefois, GoldenEye se distingue par l'utilisation de techniques de chiffrement et d'obfuscation qui lui permettent de dissimuler ses requêtes et de contourner les systèmes de sécurité et de détection classiques, rendant ainsi l'attaque plus sophistiquée et plus difficile à contrer. Le nom ń GoldenEye ż fait référence au célèbre antagoniste des films de James Bond, soulignant le caractère élaboré et redoutable de cette attaque. Par ailleurs, Slowloris constitue un autre outil d'attaque DoS qui adopte une approche différente pour épuiser les ressources du serveur cible. Il fonctionne en établissant simultanément de multiples connexions HTTP avec le serveur et en envoyant des requêtes incomplètes, qu'il ne termine jamais. Ce mécanisme maintient les connexions ouvertes indéfiniment, monopolise progressivement les ressources du serveur et finit par empêcher le traitement des requêtes légitimes. Le nom ń Slowloris ż provient d'un primate connu pour sa lenteur, en analogie avec la méthode de l'attaque qui épuise les ressources du serveur de manière progressive et discrète.

1.3.2 Déni de service distribué (DDoS)

L'attaque par déni de service distribué (DDoS) constitue une tentative malveillante visant à perturber le trafic normal d'un serveur, d'un service ou d'un réseau cible en le submergeant par un volume massif de trafic provenant d'Internet [21]. Contrairement aux attaques DoS classiques, les attaques DDoS exploitent des réseaux entiers d'ordinateurs et de dispositifs infectés, appelés botnets, qui sont contrôlés à distance par l'attaquant. Ce dernier transmet des instructions aux différentes machines zombifiées afin qu'elles envoient simultanément des requêtes vers l'adresse IP de la cible, inondant ainsi le serveur de sollicitations massives et simultanées. Cette surcharge entraîne une saturation des ressources du serveur ou du réseau visé, provoquant ainsi une interruption de service pour les utilisateurs légitimes, incapables d'accéder aux ressources sollicitées (voir Figure 1.3). L'impact de ces attaques peut être particulièrement dévastateur pour les entreprises et les infrastructures critiques, causant des pertes économiques importantes et des atteintes à la réputation. Parmi les multiples variantes d'attaques DDoS existantes, deux types sont particulièrement répandus et couramment employés, comme nous le verrons ci-après.

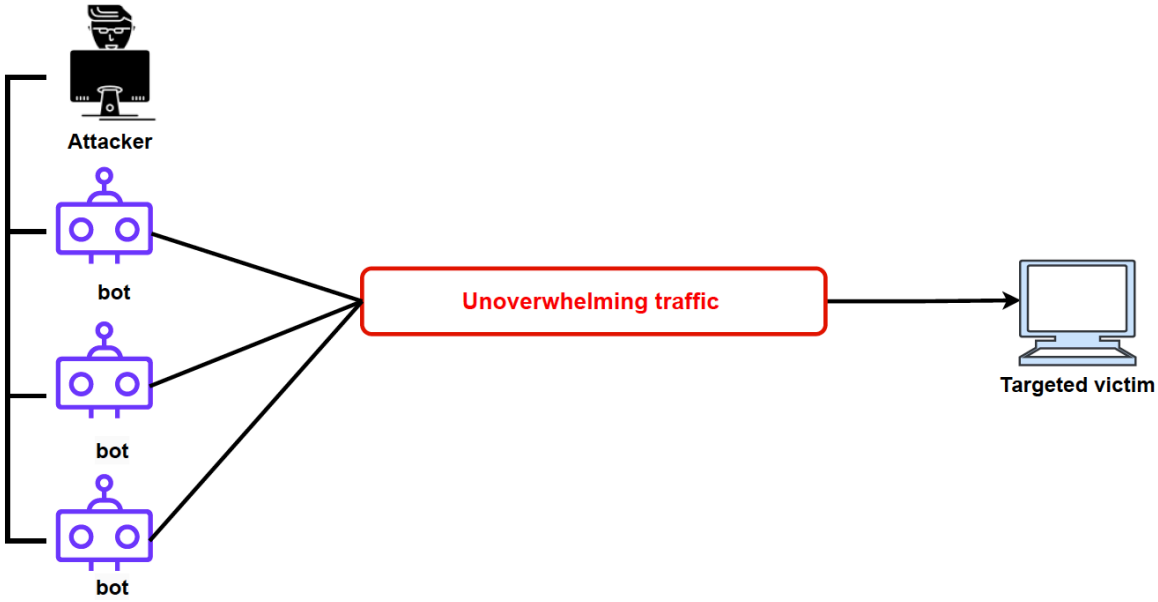


Figure 1.3 – *Attaque DDoS.*

1.3.2.1 HOIC

HOIC, acronyme de High Orbit Ion Cannon, est un outil d'attaque par déni de service distribué (DDoS) qui fonctionne en générant un très grand nombre de requêtes HTTP de type GET ou POST dans le but de saturer un serveur ou un réseau cible [22]. Cet outil est particulièrement redouté pour sa capacité à orchestrer des attaques hautement coordonnées et massives, permettant de produire des volumes de trafic extrêmement élevés en un laps de temps réduit. Grâce à cette intensité de sollicitation, le serveur ciblé est rapidement submergé, ce qui perturbe ou bloque totalement l'accès aux services pour les utilisateurs légitimes. De plus, HOIC complique considérablement l'identification de la source de l'attaque en exploitant des mécanismes de distribution et de camouflage du trafic, ce qui rend les efforts de défense et de remédiation d'autant plus complexes. Sa simplicité d'utilisation et son efficacité en ont fait un des outils les plus largement utilisés dans les attaques DDoS modernes.

1.3.2.2 LOIC

LOIC, acronyme de Low Orbit Ion Cannon, est également un outil d'attaque par déni de service distribué (DDoS) qui vise à saturer un serveur ou un réseau cible en générant un flux massif de trafic [22]. Contrairement à HOIC, LOIC est capable de mener à la fois des attaques par inondation HTTP (HTTP flood), similaires à celles utilisées par HOIC, et des attaques par inondation UDP (UDP flood). Les attaques HTTP flood consistent à envoyer un grand nombre de requêtes HTTP afin de saturer les capacités de traitement du serveur web, tandis que les attaques UDP flood impliquent l'envoi de volumes importants de paquets UDP vers la cible, épuisant ainsi

les ressources réseau et provoquant une dégradation, voire une interruption complète du service. La principale différence entre ces deux outils réside dans leur niveau de sophistication. HOIC est réputé pour sa capacité à réaliser des attaques plus avancées, mieux coordonnées et plus difficiles à détecter, alors que LOIC demeure un outil plus basique et plus accessible, souvent utilisé par des attaquants novices en raison de sa simplicité de configuration et de son interface utilisateur intuitive. Malgré sa relative simplicité, LOIC peut néanmoins causer des dommages significatifs lorsqu'il est utilisé de manière coordonnée à grande échelle.

1.3.3 Brute force

Une attaque par force brute constitue une méthode d'intrusion basée sur le principe d'essais successifs et systématiques visant à obtenir un accès non autorisé à un système ou à une application en testant de manière répétée différentes combinaisons de noms d'utilisateur et de mots de passe jusqu'à ce que la combinaison correcte soit identifiée [23]. L'attaquant exploite généralement des logiciels automatisés capables de générer un très grand nombre de tentatives en un laps de temps réduit, augmentant ainsi les chances de réussite de l'intrusion. Ces outils peuvent parcourir d'immenses bases de données de mots de passe courants ou générer aléatoirement des combinaisons de caractères, rendant la méthode efficace contre des systèmes disposant de mesures de sécurité faibles ou de mots de passe insuffisamment complexes. Ce type d'attaque, bien que relativement simple dans sa conception, peut représenter une menace sérieuse si les mécanismes de défense tels que les verrouillages de compte, les délais entre tentatives ou l'utilisation de systèmes de détection d'intrusions ne sont pas correctement mis en uvre. La Figure 1.4 illustre le fonctionnement d'une attaque par force brute.

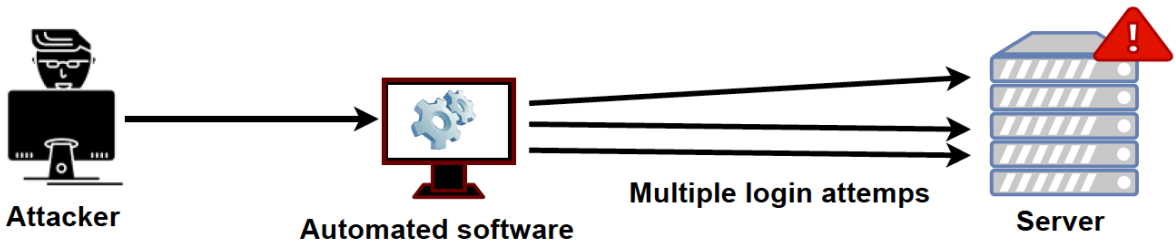


Figure 1.4 *– Attaque Brute Force.*

Quatre types différents d'attaques par force brute sont énumérés ci-dessous :

1.3.3.1 FTP Brute Force

Le File Transfer Protocol (FTP) est un protocole réseau largement utilisé pour le transfert de fichiers entre un client et un serveur au sein d'une architecture de

type client-serveur [24]. Grâce à un client FTP, les utilisateurs peuvent établir une connexion avec le serveur afin d'envoyer ou de recevoir des fichiers de manière bidirectionnelle. L'authentification des utilisateurs s'effectue généralement à l'aide d'un identifiant (nom d'utilisateur) et d'un mot de passe. Toutefois, ces informations d'identification sont souvent transmises en clair sur le réseau, ce qui expose les communications aux risques d'interception et de compromission par des tiers malveillants. En l'absence de mécanismes de sécurisation supplémentaires, tels que le chiffrement SSL/TLS utilisé dans les variantes sécurisées comme FTPS ou SFTP, le protocole FTP classique présente ainsi des vulnérabilités importantes. Par ailleurs, certains serveurs peuvent autoriser des connexions anonymes, permettant à tout utilisateur d'accéder à des fichiers publics sans authentification préalable, ce qui nécessite une gestion rigoureuse des droits d'accès pour éviter toute divulgation accidentelle de données sensibles.

1.3.3.2 SSH Brute Force

Ce type d'attaque vise spécifiquement le protocole Secure Shell (SSH), couramment utilisé pour accéder à distance aux serveurs et en assurer la gestion de manière sécurisée [25]. L'attaquant procède en testant successivement différentes combinaisons de noms d'utilisateur et de mots de passe, dans le but de trouver les identifiants corrects permettant de se connecter avec succès au serveur SSH ciblé. Une fois l'accès obtenu, l'attaquant dispose alors d'un contrôle privilégié sur le système distant, ce qui lui permet d'exécuter diverses actions malveillantes telles que le vol de données confidentielles, la modification de fichiers critiques, l'installation de logiciels malveillants (malwares) ou encore la création de portes dérobées afin de maintenir un accès clandestin au serveur. Bien que le protocole SSH offre intrinsèquement des mécanismes de sécurisation par chiffrement des communications, la robustesse des identifiants d'accès et la mise en place de mesures de protection complémentaires, comme l'utilisation de clés d'authentification ou de systèmes de détection d'intrusion, demeurent essentielles pour prévenir ce type d'attaque par force brute ciblant l'authentification SSH.

1.3.3.3 Web Brute Force

Ce type d'attaque cible spécifiquement les systèmes d'authentification en ligne accessibles via des interfaces web, tels que ceux utilisés pour les services bancaires en ligne, les comptes de messagerie électronique ou tout autre service nécessitant une connexion utilisateur [26]. L'attaquant emploie des outils automatisés capables de générer et de tester un grand nombre de combinaisons de noms d'utilisateur et de mots de passe, dans le but de découvrir les identifiants valides permettant l'accès au compte ciblé. Cette automatisation permet d'accélérer considérablement le processus de recherche par rapport à une tentative manuelle, augmentant ainsi les probabilités de succès, surtout lorsque les utilisateurs emploient des mots de passe faibles ou

courants. Une fois l'accès au compte compromis, l'attaquant peut alors procéder au vol d'informations sensibles telles que des données personnelles, financières ou professionnelles, ou encore mener diverses actions malveillantes, notamment la fraude financière, l'usurpation d'identité, la compromission d'autres comptes associés ou l'installation de logiciels malveillants à des fins d'espionnage ou de sabotage. Ce type d'attaque souligne l'importance de pratiques rigoureuses en matière de gestion des mots de passe et de l'adoption de mécanismes de sécurité supplémentaires tels que l'authentification multifacteur pour renforcer la protection des systèmes d'accès en ligne.

1.3.3.4 XSS Brute Force

Ce type d'attaque vise les applications web présentant des vulnérabilités de type cross-site scripting (XSS), c'est-à-dire des failles permettant l'injection de code malveillant dans les pages web dynamiques [27]. L'attaquant insère du contenu scripté, souvent en JavaScript, directement dans les champs de saisie ou les paramètres de l'application vulnérable. Lorsque la victime consulte la page compromise, le code malveillant s'exécute automatiquement dans le navigateur de l'utilisateur à son insu. Ce script peut alors intercepter des informations sensibles telles que l'identifiant de session, les identifiants de connexion (noms d'utilisateur et mots de passe), ou encore d'autres données confidentielles saisies ou stockées sur le navigateur. L'exploitation de ces informations permet ensuite à l'attaquant de prendre le contrôle du compte utilisateur, d'usurper son identité, ou de mener d'autres actions malveillantes sur la plateforme ciblée. Les attaques XSS représentent une menace sérieuse pour la sécurité des applications web, d'autant plus qu'elles peuvent être difficilement détectables pour l'utilisateur final et qu'elles exploitent directement la confiance accordée par le navigateur à l'application web visitée.

1.3.4 SQL Injection

Le Structured Query Language (SQL) est au cur d'une catégorie d'attaques par injection de code, connue sous le nom d'injection SQL [28], qui cible spécifiquement les bases de données relationnelles en insérant des instructions SQL malveillantes dans les champs de saisie de données des applications web (voir Figure 1.5). L'attaquant exploite l'absence de validation ou de filtrage adéquat des données saisies par l'utilisateur pour injecter directement du code SQL dans les requêtes exécutées par le serveur. Grâce à cette manipulation, l'attaquant peut interroger la base de données de manière non autorisée afin d'en extraire des informations sensibles telles que des données personnelles, des identifiants de connexion, ou des informations financières. Au-delà de l'exfiltration de données, les attaques par injection SQL permettent également de modifier ou de supprimer des données existantes, voire de détruire l'intégralité de la base de données dans les cas les plus critiques. Ce type d'attaque est

fréquemment perpétré via des interfaces web accessibles sur Internet, en envoyant des requêtes malicieuses à des points de terminaison d'API exposés par les sites ou les services vulnérables. L'injection SQL reste l'une des vulnérabilités les plus critiques en matière de cybersécurité des applications web et figure régulièrement parmi les premières positions dans les classements des failles identifiées par l'OWASP.

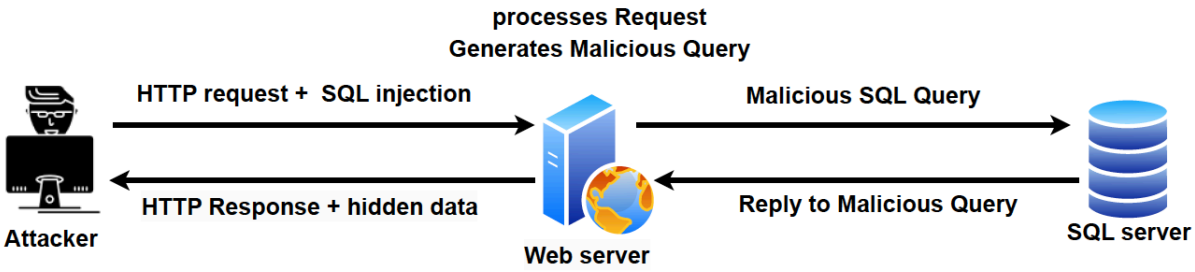


Figure 1.5 – *Attaque SQL Injection.*

1.3.5 Infiltration

L'infiltration désigne le processus par lequel un attaquant parvient à obtenir un accès non autorisé à un système ou à un réseau informatique, généralement à des fins malveillantes [29]. Pour mener à bien une infiltration, les cybercriminels recourent à une diversité de techniques, incluant l'exploitation de vulnérabilités logicielles ou matérielles, le recours à des méthodes d'ingénierie sociale afin de duper les utilisateurs et les inciter à divulguer des informations sensibles ou à exécuter du code malveillant, ainsi que la violation physique des dispositifs de sécurité. Une forme courante d'infiltration consiste à exploiter des vulnérabilités depuis l'intérieur même du réseau ciblé. Dans ce cas de figure, l'attaquant peut, par exemple, envoyer un fichier malveillant par courrier électronique à une victime, exploitant ensuite une vulnérabilité d'application lors de l'ouverture du fichier. Une fois l'attaque réussie, un programme de type backdoor est installé sur l'ordinateur de la victime, offrant à l'attaquant un point d'accès permanent au réseau interne. Grâce à cette porte dérobée, l'attaquant peut alors cartographier le réseau local, rechercher d'autres systèmes vulnérables et les compromettre à leur tour. Cette capacité à se déplacer latéralement à l'intérieur d'un réseau après une compromission initiale constitue l'un des aspects les plus redoutables de l'infiltration moderne.

1.3.6 Botnet

Un botnet désigne un réseau de dispositifs compromis, contrôlés à distance par une entité unique ou un attaquant, et utilisés pour mener diverses activités malveillantes à grande échelle [30]. Les appareils infectés, souvent appelés bots ou zombies, sont contaminés par des logiciels malveillants qui permettent au botmaster (le maître du

botnet) d'en prendre le contrôle à distance. Une fois activé, le botnet peut être mobilisé pour exécuter plusieurs types d'attaques, notamment des attaques par déni de service distribué (DDoS), l'envoi massif de courriels indésirables (spamming), des campagnes de phishing, des attaques par bourrage d'identifiants (credential stuffing), ainsi que d'autres opérations de cybercriminalité. Parmi les malwares de botnet les plus connus figure Zeus, un cheval de Troie ciblant les systèmes d'exploitation Microsoft Windows. Ce malware est fréquemment utilisé pour des activités criminelles telles que le vol de données bancaires sensibles et l'installation de ransomwares. La propagation de Zeus s'effectue souvent via des téléchargements invisibles (drive-by downloads) ou des campagnes de phishing sophistiquées. Un autre botnet couramment utilisé est Ares, un botnet open source offrant au botmaster une large gamme de fonctionnalités offensives, incluant l'exécution de commandes shell à distance, la persistance sur les systèmes infectés, le transfert de fichiers (téléchargement et envoi), la capture d'écran ainsi que l'enregistrement des frappes clavier (keylogging). Ces botnets constituent aujourd'hui des instruments puissants de la cybercriminalité organisée, capables de causer des dommages considérables à l'échelle mondiale.

En résumé, les menaces à la sécurité des systèmes informatiques peuvent causer des dommages considérables aux réseaux, se traduisant par des violations de données, des interruptions de service, des pertes de productivité et des pertes financières importantes [31]. La protection contre ces menaces nécessite une approche globale et multidimensionnelle de la sécurité des réseaux, reposant sur la combinaison de mesures techniques, organisationnelles et humaines. Il est essentiel d'effectuer régulièrement des mises à jour de sécurité sur les logiciels et les équipements matériels afin de corriger les vulnérabilités nouvellement découvertes. L'utilisation de mots de passe robustes et uniques pour chaque utilisateur constitue également une barrière fondamentale contre les intrusions. Par ailleurs, la formation et la sensibilisation des utilisateurs jouent un rôle crucial pour réduire l'efficacité des attaques d'ingénierie sociale, qui exploitent la naïveté ou l'inattention des individus. En complément de ces mesures préventives, l'intégration de dispositifs de sécurité tels que les pare-feu, les logiciels antivirus, les systèmes de détection et de prévention des intrusions (IDS/IPS), ainsi que des mécanismes de contrôle d'accès rigoureux, permet de détecter rapidement les attaques potentielles et d'en atténuer les effets avant qu'elles n'entraînent des conséquences majeures.

1.4 Mesures de sécurité des réseaux

La sécurisation d'un réseau informatique peut être réalisée à l'aide de plusieurs méthodes complémentaires visant à protéger l'intégrité, la confidentialité et la disponibilité des données et des ressources système contre les menaces internes et externes [31]. Parmi les approches les plus couramment mises en uvre figurent l'utilisa-

tion de pare-feu pour contrôler et filtrer le trafic réseau entrant et sortant, les systèmes de détection et de prévention des intrusions (IDS/IPS) permettant d'identifier et de neutraliser les activités suspectes, ainsi que le chiffrement des communications afin de garantir la confidentialité des échanges de données. D'autres mesures essentielles incluent la gestion rigoureuse des accès utilisateurs via des politiques d'authentification et d'autorisation robustes, la segmentation des réseaux pour limiter la propagation des attaques, et la mise en uvre de programmes de sensibilisation à la cybersécurité destinés aux utilisateurs afin de réduire les risques liés à l'ingénierie sociale. L'application régulière des correctifs de sécurité sur les systèmes et les applications est également indispensable pour remédier aux vulnérabilités connues et ainsi renforcer la résilience globale du réseau face aux cybermenaces.

1.4.1 Pare-feux

Les pare-feu constituent un outil essentiel dans la sécurisation des réseaux informatiques contre les accès non autorisés et les attaques de logiciels malveillants [32]. Leur fonction principale est de surveiller le trafic entrant et sortant du réseau en appliquant des règles de filtrage prédéfinies, permettant ainsi de bloquer les connexions provenant de sources qui ne répondent pas à certains critères spécifiques, tels que l'adresse IP, le numéro de port ou le protocole utilisé. Grâce à leur capacité à contrôler le flux de données en temps réel, les pare-feu permettent de prévenir de nombreuses tentatives d'intrusion et de limiter la propagation de menaces au sein du réseau. Leur déploiement est généralement relativement simple, et leur administration peut être centralisée, facilitant ainsi la gestion continue de la sécurité. Toutefois, malgré leur efficacité pour filtrer un grand nombre de menaces courantes, les pare-feu présentent certaines limites, notamment dans la détection des attaques sophistiquées de type zero-day, des menaces internes provenant d'utilisateurs malveillants autorisés, ou encore des attaques dissimulées dans des flux de trafic chiffrés. C'est pourquoi ils doivent être complétés par d'autres mécanismes de défense avancés pour assurer une protection globale et approfondie. La Figure 1.6 illustre le fonctionnement général d'un pare-feu.

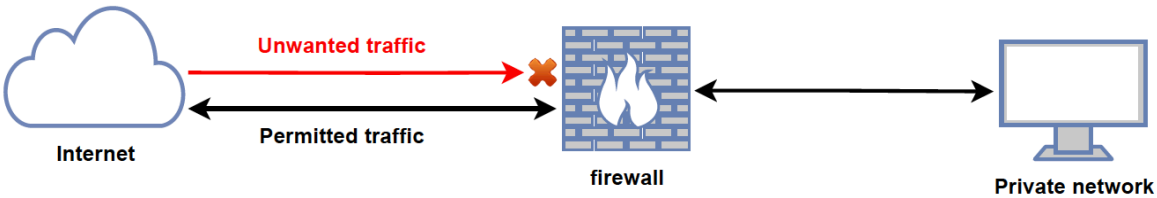


Figure 1.6 – *Pare-feu.*

Il existe plusieurs types de pare-feu [33] couramment utilisés dans les réseaux informatiques :

1.4.1.1 Pare-feu à filtrage de paquets

Ce type de pare-feu, appelé pare-feu à filtrage de paquets, fonctionne au niveau de la couche réseau et examine les paquets de données en filtrant le trafic entrant et sortant sur la base de règles prédéfinies, telles que les adresses IP source et destination, les numéros de port et le type de protocole utilisé [32]. Grâce à leur simplicité de mise en uvre et à leur faible impact sur les performances du réseau, ces pare-feux sont largement adoptés pour assurer une première ligne de défense dans de nombreuses infrastructures informatiques. Leur coût modéré et leur facilité de gestion les rendent attractifs, en particulier dans des contextes nécessitant des solutions de sécurité basiques et rapides à déployer. Toutefois, en raison de leur fonctionnement limité à l'analyse statique des en-têtes de paquets, ils présentent des vulnérabilités face à certaines attaques plus sophistiquées, telles que le spoofing d'adresses IP, qui consiste à usurper l'identité d'une source de confiance, ou encore les attaques par déni de service (DoS), qui visent à saturer les ressources du réseau. Par conséquent, bien que les pare-feux à filtrage de paquets offrent une protection élémentaire et économiquement avantageuse, ils ne suffisent pas à eux seuls pour faire face aux menaces modernes et doivent impérativement être intégrés à une stratégie de sécurité plus globale, comprenant des solutions complémentaires telles que les systèmes de détection d'intrusion, les pare-feux applicatifs et les solutions de surveillance proactive.

1.4.1.2 Pare-feu à inspection dynamique

Ce type de pare-feu, appelé pare-feu à inspection dynamique des états (stateful inspection firewall), surveille et enregistre l'état des connexions réseau en cours afin de prendre des décisions de filtrage basées sur des critères tels que l'état de la connexion, le numéro de port et le protocole utilisé [32]. Contrairement aux pare-feux à simple filtrage de paquets qui analysent chaque paquet de manière indépendante, le pare-feu à inspection dynamique suit l'intégralité de la session, depuis l'établissement de la connexion jusqu'à sa fermeture, en conservant des informations contextuelles sur les échanges précédents. Ainsi, les décisions de filtrage sont prises en tenant compte à la fois des règles définies par l'administrateur et du contexte d'une communication donnée, ce qui permet de mieux distinguer le trafic légitime des flux malveillants. Grâce à cette capacité de suivi des connexions, ces pare-feux sont également capables de détecter et de bloquer certains types d'attaques spécifiques, telles que les attaques par SYN flood, qui visent à submerger un serveur en multipliant les tentatives de connexions incomplètes. En maintenant une table d'état des connexions autorisées, le pare-feu peut ainsi refuser les paquets qui ne correspondent pas à des sessions valides, renforçant significativement la sécurité du réseau contre de nombreuses menaces basées sur la manipulation des protocoles de communication.

1.4.1.3 Pare-feu de nouvelle génération (NGFW)

Le pare-feu de nouvelle génération (NGFW, pour Next Generation Firewall) représente une évolution avancée des pare-feux traditionnels, intégrant des fonctionnalités supplémentaires qui permettent d'offrir une protection beaucoup plus complète face aux menaces actuelles [32]. Outre les fonctions classiques de filtrage des paquets et de contrôle des connexions, les NGFW embarquent des systèmes de prévention des intrusions (IPS), des mécanismes de visibilité et de contrôle des applications, des dispositifs d'inspection des flux chiffrés SSL/TLS (SSL inspection), ainsi que des solutions de protection avancée contre les malwares. Grâce à l'inspection approfondie des paquets (deep packet inspection), ces pare-feux sont capables d'analyser en détail le contenu, le contexte, le comportement et l'utilisateur à l'origine du trafic, permettant ainsi de bloquer de manière ciblée les flux suspects ou malveillants selon les politiques de sécurité définies. Cette granularité de contrôle offre une visibilité fine sur les applications et les utilisateurs présents sur le réseau, rendant possible l'application de règles très spécifiques. Les NGFW sont également capables de détecter et de contrer des attaques sophistiquées telles que les vulnérabilités zero-day et les menaces persistantes avancées (Advanced Persistent Threats APT), en s'appuyant sur l'analyse en temps réel des flux de données et sur l'intégration de bases de renseignement sur les menaces (threat intelligence feeds). Par leur capacité à combiner plusieurs couches de défense en un seul dispositif, les NGFW offrent un niveau de sécurité nettement supérieur à celui des pare-feux classiques, ce qui en fait aujourd'hui un choix privilégié pour la protection des infrastructures réseau modernes, complexes et hautement exposées aux cyberattaques.

1.4.2 Réseaux privés virtuels (VPN)

Les réseaux privés virtuels (Virtual Private Networks VPN) permettent d'assurer un accès distant sécurisé à un réseau en créant un tunnel chiffré à travers Internet, garantissant ainsi la confidentialité et l'intégrité des données échangées entre l'utilisateur distant et le réseau interne de l'organisation (voir Figure 1.7) [32]. Cette technologie est particulièrement prisée par les employés en télétravail ou en déplacement, qui doivent accéder aux ressources de l'entreprise depuis l'extérieur des locaux. En encapsulant les paquets de données dans un canal sécurisé et en appliquant des algorithmes de chiffrement robustes, le VPN protège les communications contre les interceptions, l'espionnage et les tentatives de manipulation de données, même lorsque l'utilisateur utilise des réseaux publics ou non sécurisés. Outre la confidentialité, les VPN assurent également l'authentification des utilisateurs et la vérification de l'intégrité des données transmises. Leur déploiement s'avère aujourd'hui essentiel pour maintenir la continuité des opérations dans un contexte de mobilité professionnelle croissante, tout en limitant les risques d'exposition aux cyberattaques lors des

connexions à distance aux systèmes critiques de l'entreprise.

Figure 1.7 – *Réseau VPN.*

Il existe plusieurs types de VPN vpn, notamment :

1.4.2.1 Accès à distance

Le VPN d'accès à distance (Remote Access VPN) permet de connecter de manière sécurisée un dispositif situé en dehors des locaux de l'entreprise à un réseau privé via Internet [32]. Ce type de VPN est largement utilisé par les employés ayant besoin d'accéder aux ressources de l'entreprise depuis l'extérieur, que ce soit depuis leur domicile, en déplacement ou en télétravail. Grâce à la création d'un tunnel chiffré, le VPN d'accès à distance garantit la confidentialité des données échangées et protège les communications contre les interceptions ou les attaques de type man-in-the-middle. Les évolutions récentes des technologies VPN ont permis l'intégration de mécanismes de contrôle de sécurité des points d'accès (endpoints) avant l'établissement de la connexion, assurant ainsi que les appareils des utilisateurs respectent les politiques de sécurité en vigueur (mises à jour logicielles, présence d'antivirus, pare-feu actif, etc.) avant d'autoriser l'accès aux ressources sensibles du réseau privé. Cette approche contribue à préserver non seulement la confidentialité, mais également l'intégrité et la disponibilité des données critiques de l'entreprise, tout en permettant aux collaborateurs d'exercer leurs activités professionnelles à distance en toute sécurité.

1.4.2.2 Site à site

Le VPN de site à site (Site-to-Site VPN) est un type de réseau privé virtuel qui permet d'établir une connexion sécurisée entre le siège d'une entreprise et ses succursales via Internet, en reliant de manière chiffrée deux ou plusieurs réseaux distincts [32]. Ce type de VPN est couramment utilisé pour assurer la connectivité des bureaux distants, faciliter les migrations vers le cloud, ou encore garantir la continuité des opérations dans le cadre de plans de reprise après sinistre. Contrairement aux connexions directes, souvent impraticables en raison de la distance ou des contraintes d'infrastructure, le VPN de site à site repose sur des équipements dédiés (tels que des routeurs VPN ou des passerelles de sécurité) permettant d'établir et de maintenir la liaison de manière automatisée et sécurisée. Cette architecture, également

désignée sous le terme de connexion réseau à réseau (network-to-network access), offre aux organisations la possibilité d'interconnecter plusieurs réseaux d'entreprise, ou encore de relier un réseau d'entreprise à celui d'un fournisseur de services cloud, tout en préservant la confidentialité, l'intégrité et la disponibilité des données sensibles échangées entre les différentes entités. Le VPN de site à site constitue ainsi une solution robuste et scalable pour garantir une interconnexion sécurisée et fiable entre des infrastructures géographiquement dispersées.

1.4.3 Cryptage

Le chiffrement constitue une technique de protection particulièrement efficace pour garantir la confidentialité et l'intégrité des données en transformant un texte en clair en une forme codée, illisible par des parties non autorisées [32]. Ce procédé de sécurisation est largement utilisé dans divers contextes, tels que la communication sécurisée par courrier électronique, la transmission de données sensibles via des réseaux publics, ou encore le stockage sécurisé d'informations critiques sur des systèmes de stockage locaux ou dans le cloud. En rendant les données inexploitables sans la clé de déchiffrement appropriée, le chiffrement protège les informations contre les interceptions, les vols et les manipulations malveillantes. Toutefois, sa mise en uvre peut engendrer certains impacts sur les performances des systèmes, notamment en termes de latence et de consommation de ressources, et nécessite des compétences de gestion avancées en raison de la complexité des mécanismes de gestion des clés, de la conformité réglementaire et du cycle de vie des certificats. Malgré ces contraintes, les avantages considérables qu'offre le chiffrement en matière de protection des informations sensibles en font un outil incontournable dans toute stratégie de sécurité informatique moderne. Il existe plusieurs types de chiffrement adaptés aux différents besoins de sécurité, notamment :

1.4.3.1 Cryptage symétrique et asymétrique

Le chiffrement symétrique, également appelé chiffrement à clé partagée (shared-secret encryption), repose sur l'utilisation d'une seule et même clé pour assurer à la fois le chiffrement et le déchiffrement des données [34]. Cette approche présente l'avantage d'être particulièrement rapide et efficace sur le plan computationnel, ce qui la rend bien adaptée au traitement de grands volumes de données. Toutefois, la sécurité du chiffrement symétrique dépend entièrement du secret de la clé : si celle-ci venait à être compromise ou interceptée, l'ensemble des données chiffrées deviendrait accessible à un attaquant. En parallèle, le chiffrement asymétrique, connu sous le nom de chiffrement à clé publique (public-key encryption), utilise une paire de clés distinctes : une clé publique, librement distribuée, permettant de chiffrer les données, et une clé privée, conservée secrète par le destinataire, servant à les déchiffrer. Bien que le chiffrement asymétrique soit généralement plus lent que son homologue symétrique

en raison de la complexité des calculs mathématiques impliqués, il offre un niveau de sécurité renforcé puisque la clé privée reste strictement confidentielle, rendant extrêmement difficile pour un attaquant d'accéder aux données chiffrées sans disposer de cette clé. Ce modèle constitue ainsi la base de nombreuses applications de sécurité moderne, telles que les protocoles SSL/TLS, la signature électronique, ou encore les infrastructures à clé publique (PKI).

1.4.3.2 Hachage

Le hachage est un processus de chiffrement à sens unique qui transforme un texte en clair en une chaîne de caractères de longueur fixe, appelée empreinte ou valeur de hachage [35]. Pour une même entrée de texte en clair, le processus génère toujours la même empreinte, ce qui permet de vérifier l'intégrité des données sans avoir besoin d'en connaître le contenu original. Toutefois, en raison de la nature irréversible de cette opération, il est pratiquement impossible, d'un point de vue computationnel, de reconstituer les données initiales à partir de leur empreinte de hachage, sauf en cas d'attaque par force brute ou par utilisation de tables arc-en-ciel pré-calculées. Le hachage est ainsi largement utilisé dans de nombreux domaines de la sécurité informatique, notamment pour le stockage sécurisé des mots de passe, la vérification de l'intégrité des fichiers, ou encore la signature numérique. Parmi les algorithmes de hachage les plus couramment employés figurent MD5 (désormais obsolète en raison de ses vulnérabilités), SHA (dans ses différentes variantes telles que SHA-1, SHA-256, SHA-512), ainsi que des algorithmes plus robustes et adaptés à la protection des mots de passe tels que bcrypt et scrypt, qui intègrent des mécanismes de ralentissement computationnel afin de rendre plus difficiles les attaques par force brute sur les mots de passe chiffrés.

1.4.4 Contrôle d'accès

Le contrôle d'accès constitue une stratégie essentielle dans la conception des réseaux visant à restreindre l'accès aux ressources et aux infrastructures uniquement aux dispositifs terminaux conformes, authentifiés et de confiance, tout en empêchant l'accès non autorisé et les menaces potentielles [35]. Ce mécanisme repose sur la mise en uvre de processus d'authentification, tels que l'utilisation de mots de passe robustes, d'identifiants utilisateurs uniques et de procédures d'authentification forte, permettant de vérifier l'identité des utilisateurs avant de leur accorder l'accès aux ressources du réseau. En limitant l'accès aux données et aux ressources sensibles exclusivement aux individus autorisés, le contrôle d'accès joue un rôle déterminant dans la prévention des intrusions et des violations de données. Il est couramment utilisé pour faire respecter les politiques de mots de passe, gérer les autorisations d'accès selon les rôles des utilisateurs, et assurer la traçabilité des activités réalisées sur le réseau. Toutefois, la mise en uvre efficace du contrôle d'accès peut s'avérer complexe,

nécessitant une gestion rigoureuse des droits d'accès et une surveillance constante afin d'éviter les configurations erronées ou obsolètes qui pourraient créer des failles de sécurité. Par ailleurs, un contrôle d'accès mal configuré peut générer un faux sentiment de sécurité, laissant croire que le système est protégé alors qu'il reste vulnérable à certaines formes d'attaques internes ou externes.

1.4.5 Systèmes de gestion des informations et des événements de sécurité (SIEM)

Le SIEM (Security Information and Event Management) est un type de logiciel qui assure l'analyse en temps réel des alertes de sécurité générées par les équipements réseau et les applications [36]. Les systèmes SIEM combinent les fonctionnalités de Security Information Management (SIM), axées sur la collecte, la centralisation et la conservation des journaux de sécurité, et de Security Event Management (SEM), spécialisées dans la corrélation et l'analyse des événements en temps réel, afin de fournir une vision globale et cohérente de l'état de sécurité d'une organisation. Ces systèmes agrègent des données de sécurité provenant de multiples sources hétérogènes telles que les pare-feux, les systèmes de détection d'intrusion (IDS), les serveurs, les équipements réseau et les applications métiers critiques. Grâce à l'utilisation de moteurs analytiques sophistiqués et d'algorithmes d'apprentissage automatique (machine learning), les SIEM sont capables de détecter des schémas d'activité anormaux ou des corrélations d'événements pouvant signaler une violation de sécurité ou une menace émergente. Lorsqu'un incident potentiel est identifié, le système SIEM génère une alerte destinée aux analystes ou aux administrateurs de sécurité en vue d'une investigation approfondie. Certains SIEM modernes intègrent également des fonctionnalités d'automatisation de la réponse aux incidents, capables de déclencher automatiquement des actions correctives telles que le blocage d'une adresse IP malveillante, l'isolement d'un segment réseau compromis ou la mise hors service d'un système infecté, contribuant ainsi à limiter rapidement les impacts des attaques.

Voici quelques-unes des principales caractéristiques des systèmes SIEM :

1. Collecte et gestion des logs : Collecte et stockage de logs provenant de diverses sources dans un lieu central à des fins d'analyse et de conservation.
2. Corrélation d'événements en temps réel : Corrélation des événements de sécurité provenant de sources multiples afin d'obtenir une vue d'ensemble des menaces de sécurité.
3. Surveillance de l'activité des utilisateurs : Surveillance de l'activité des utilisateurs pour détecter les accès non autorisés ou les comportements suspects.
4. Renseignements sur les menaces : Intégration avec des flux externes de renseignements sur les menaces afin d'améliorer les capacités de détection et de réponse aux menaces.

5. Rapports et conformité : Génération de rapports à des fins de conformité et pour fournir une visibilité sur la posture de sécurité.

Parmi les systèmes SIEM les plus utilisés aujourd'hui, citons IBM QRadar, Splunk Enterprise Security, LogRhythm et McAfee Enterprise Security Manager.

1.4.6 Zero Trust Network Access

Le Zero Trust Network Access (ZTNA) est un modèle de cybersécurité qui repose sur le principe du ń Never trust, always verifyż (ne jamais faire confiance, toujours verifier) [37]. Contrairement aux VPN traditionnels qui accordent un accès complet au réseau une fois authentifié, le ZTNA ne donne accès qu'aux ressources spécifiques, après vérification stricte de l'identité et du contexte (appareil, localisation, comportement).

Le ZTNA agit comme un courtier d'accès (access broker) entre l'utilisateur et l'application [38] :

- Authentification : Avant chaque accès, l'identité est validée via des méthodes comme l'authentification multifactorielle (MFA) ou les certificats numériques.
- Controle basé sur des politiques : Les règles d'accès sont définies par l'organisation (par exemple, un employé peut accéder uniquement à l'application RH, mais pas aux serveurs financiers).
- Segmentation et micro-segmentation : Le réseau est découpé en segments isolés pour limiter la propagation d'éventuelles intrusions.
- Surveillance continue : Même après l'accès, les sessions sont surveillées en temps réel pour détecter tout comportement anormal.

Le ZTNA offre plusieurs bénéfices majeurs pour les organisations [39] :

- Sécurite renforcée : contrairement aux VPN, il ne donne jamais un accès global au réseau, ce qui réduit considérablement la surface d'attaque.
- Protection du cloud et du télétravail : Il s'adapte parfaitement aux environnements hybrides où les utilisateurs accèdent aux ressources depuis divers lieux et appareils.
- Réduction des risques internes : Même un employé ou appareil compromis ne peut pas se déplacer librement dans le réseau.
- Expérience utilisateur améliorée : Grâce a des accès cibles, les utilisateurs obtiennent directement ce dont ils ont besoin sans passer par des tunnels VPN lourds.
- Conformité réglementaire : En appliquant des contrôles fins et une traçabilité continue, il aide les entreprises à répondre aux exigences de conformité (ISO, GDPR, NIS2, etc.).

1.4.7 Systèmes de détection et de prévention des intrusions (IDPS)

Les systèmes de détection et de prévention des intrusions (Intrusion Detection and Prevention Systems IDPS) sont des technologies de sécurité conçues pour détecter et empêcher les accès non autorisés, les usages abusifs ainsi que diverses menaces pesant sur les systèmes informatiques et les réseaux [36]. Les IDPS fonctionnent en surveillant en temps réel les événements réseau et système, en les analysant à la recherche de signes d'activités malveillantes et en prenant des mesures préventives ou correctives afin de neutraliser ou d'atténuer les menaces identifiées. On distingue principalement deux grandes catégories d'IDPS : les IDPS basés sur le réseau (Network-based IDPS NIDPS) et les IDPS basés sur l'hôte (Host-based IDPS HIDPS). Les NIDPS surveillent le trafic réseau à la recherche d'activités suspectes, telles que des signatures d'attaques connues, des comportements anormaux ou des flux non conformes aux politiques de sécurité, et peuvent agir en bloquant ou en isolant le trafic malveillant détecté. Les HIDPS, quant à eux, opèrent directement sur les systèmes hôtes individuels (serveurs, postes de travail) et sont capables de détecter des activités malicieuses qui pourraient ne pas être visibles au niveau du réseau, telles que des modifications de fichiers systèmes critiques, des tentatives d'escalade de privilèges ou des comportements anormaux des processus. Grâce à leur complémentarité, les IDPS permettent de renforcer considérablement la posture de sécurité globale en assurant une surveillance à la fois réseau et locale, en détectant de manière proactive les attaques connues et émergentes avant qu'elles n'entraînent des conséquences graves pour l'organisation.

Voici quelques-unes des principales caractéristiques des systèmes IDPS :

1. Surveillance en temps réel des événements du réseau et du système.
2. Analyse du trafic réseau pour détecter les signes d'activité malveillante.
3. Mécanismes de réponse et de prévention automatiques pour stopper les menaces en temps réel.
4. Des capacités d'alerte et de reporting pour informer les équipes de sécurité des menaces et des violations potentielles.

En conclusion, il existe plusieurs méthodes pour sécuriser un réseau, chacune ayant ses propres forces et faiblesses. Cependant, la méthode la plus importante pour sécuriser un réseau est un IDPS, qui peut détecter les menaces et y répondre en temps réel. En combinant ces méthodes et en suivant les meilleures pratiques telles que la mise à jour régulière des logiciels et l'utilisation de mots de passe forts, les entreprises et les particuliers peuvent réduire de manière significative le risque de violation de la sécurité.

1.5 Système de détection d'intrusion

Un système de détection d'intrusion (Intrusion Detection System IDS) constitue un élément essentiel de toute stratégie de sécurité réseau complète [40]. Sa fonction principale est de surveiller en continu l'ensemble du trafic réseau, les journaux système et d'autres sources d'événements afin de détecter et de signaler aux administrateurs de sécurité toute tentative potentielle de violation de sécurité ou d'accès non autorisé à un réseau ou à un système. L'IDS est capable d'identifier une vaste gamme de menaces, incluant les infections par logiciels malveillants, les tentatives d'exploitation de vulnérabilités connues, les activités non autorisées des utilisateurs internes ainsi que les comportements suspects caractéristiques de tentatives d'intrusion. En assurant une surveillance proactive, l'IDS permet aux organisations de repérer rapidement les menaces émergentes et d'intervenir avant qu'elles ne provoquent des dégâts importants, tels que le vol de données sensibles, la perturbation des systèmes critiques ou la dégradation de la réputation de l'organisation. En l'absence de ce dispositif, les attaquants peuvent agir plus librement et compromettre les ressources de l'entreprise sans détection immédiate. Grâce à l'implémentation d'un IDS, les administrateurs de sécurité sont en mesure de localiser rapidement les incidents de sécurité, de limiter l'impact des attaques en cours et de renforcer les défenses pour prévenir les attaques futures. La Figure 1.8 illustre le fonctionnement d'un système de détection d'intrusion.

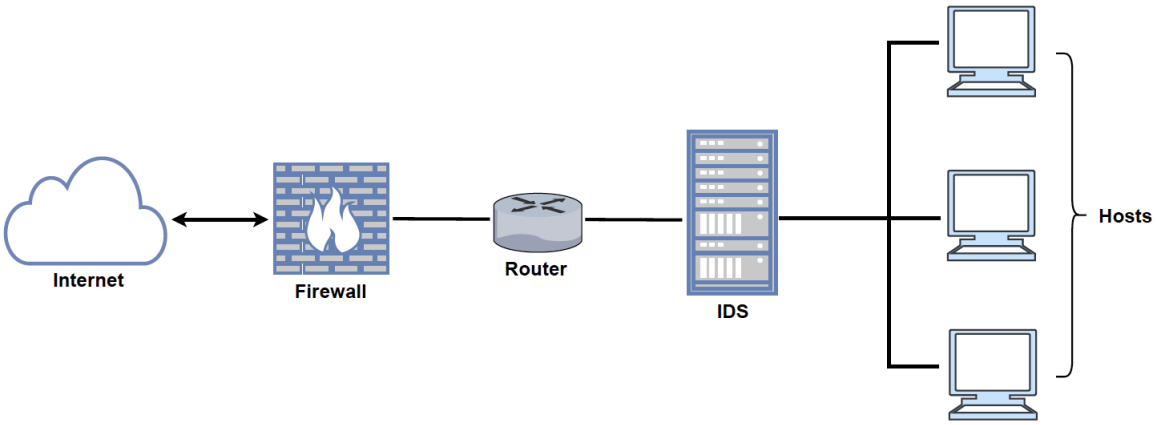


Figure 1.8 – *Système de détection d'intrusion.*

1.6 Types de systèmes de détection d'intrusion

Il existe différents types de systèmes IDS, chacun ayant ses propres forces et capacités. Les trois principaux types d'IDS sont énumérés ci-dessous.

1.6.1 IDS en réseau (NIDS)

Les systèmes de détection d'intrusion basés sur le réseau (Network-based Intrusion Detection Systems NIDS) représentent une catégorie spécifique de systèmes IDS positionnés stratégiquement au sein des infrastructures réseau afin de surveiller de manière passive le trafic circulant sur le réseau [41]. Ces systèmes peuvent être implémentés sous forme matérielle ou logicielle et sont capables de se connecter à divers supports réseau, tels qu'Ethernet ou FDDI (Fiber Distributed Data Interface). En règle générale, un NIDS est équipé de deux interfaces réseau distinctes : l'une configurée en mode promiscuité (promiscuous mode) pour écouter l'intégralité du trafic transitant sur le segment réseau surveillé, et l'autre dédiée aux fonctions de contrôle, de gestion et de génération de rapports. Cette architecture permet au NIDS d'analyser en permanence le flux de données circulant sur le réseau, en recherchant des indicateurs de compromission tels que les infections par malwares, les tentatives d'accès non autorisées, les communications anormales ou les signatures d'attaques connues. En plus de détecter les menaces, certains NIDS modernes disposent de capacités de blocage actif et de réponse automatisée, en interrompant les connexions malveillantes et en alertant les administrateurs de sécurité via des rapports détaillés, facilitant ainsi la gestion proactive des incidents de sécurité. La Figure 1.9 illustre le fonctionnement d'un système de détection d'intrusion basé sur le réseau.

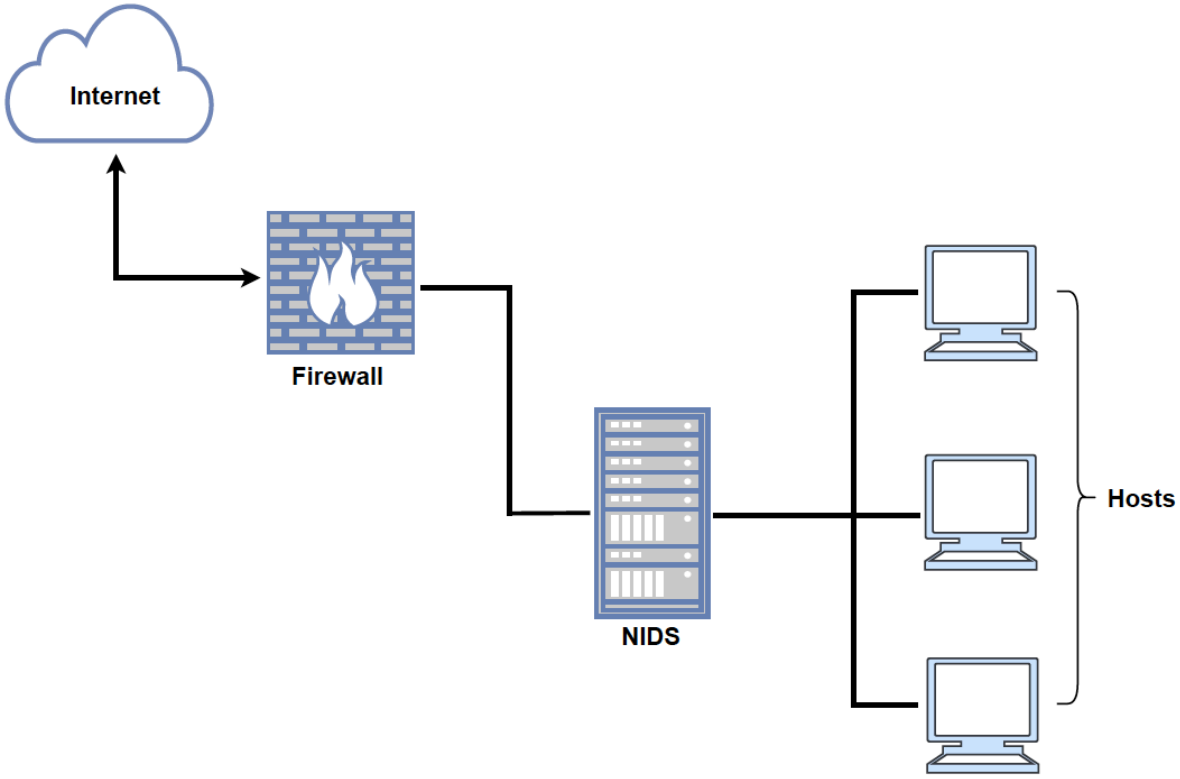


Figure 1.9 *– IDS en réseau (NIDS).*

1.6.2 IDS basé sur l'hôte (HIDS)

Le système de détection d'intrusion basé sur l'hôte (Host-based Intrusion Detection System HIDS) est une application installée directement sur une machine spécifique, chargée de surveiller en continu le système ou le réseau local afin de détecter toute activité suspecte, qu'il s'agisse d'intrusions provenant de sources externes ou de mauvaises utilisations des ressources et des données en interne [42]. Le logiciel HIDS consigne toutes les activités anormales détectées et en informe les administrateurs de sécurité, leur permettant ainsi d'identifier rapidement les anomalies et les signes d'intrusion potentielle ayant pu survenir sur le système surveillé. Ces outils examinent de manière approfondie les fichiers journaux générés par les applications et les systèmes d'exploitation, créant ainsi un historique détaillé des activités et des fonctions exécutées sur la machine. La plupart des HIDS modernes utilisent une combinaison de méthodes de détection basées sur les signatures (comparaison avec des bases de données de menaces connues) et sur les anomalies (détection de comportements inhabituels), intégrant de plus en plus des techniques d'apprentissage automatique pour affiner la reconnaissance des comportements malveillants. La capacité clé qui rend les HIDS indispensables réside dans leur fonction de détection automatisée, évitant ainsi aux administrateurs de devoir analyser manuellement d'importants volumes de journaux pour identifier les comportements suspects. En s'appuyant sur des règles et des politiques de sécurité prédéfinies, le HIDS analyse les journaux et signale automatiquement les événements ou activités pouvant indiquer une compromission potentielle. La Figure 1.10 illustre le fonctionnement d'un système de détection d'intrusion basé sur l'hôte.

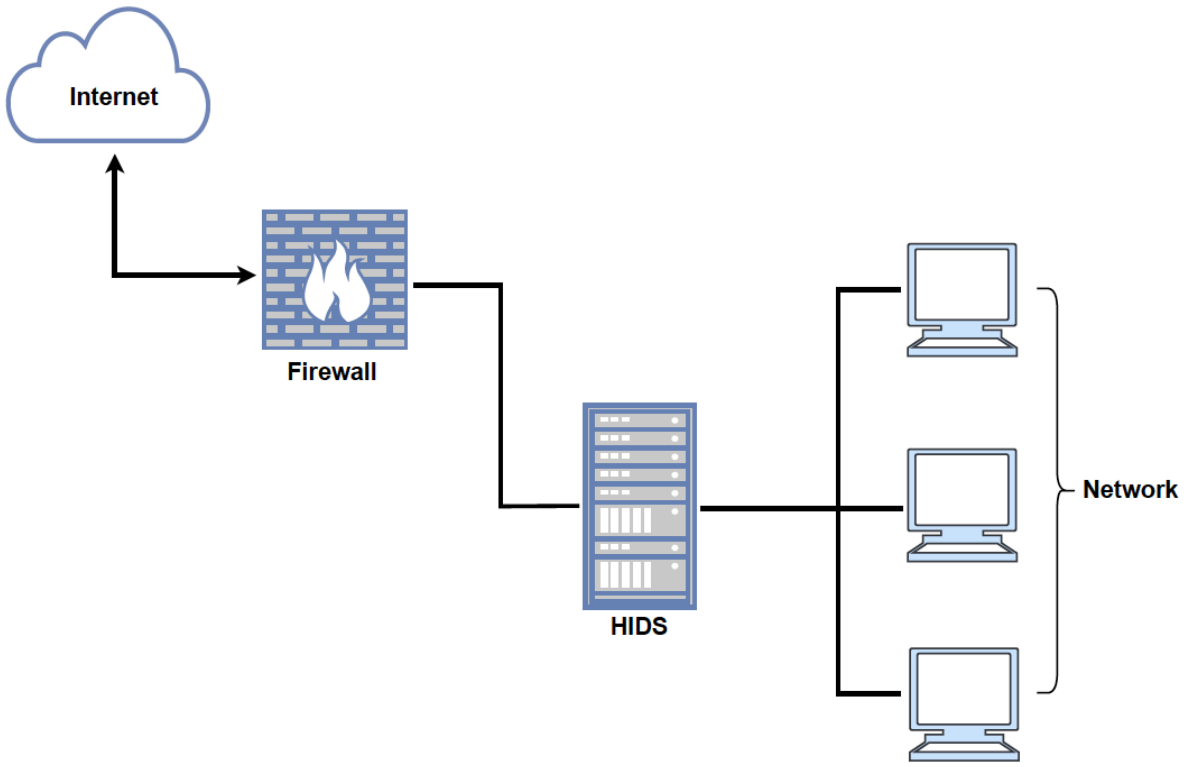


Figure 1.10 *– IDS basé sur l'hôte (HIDS).*

Le HIDS (Host-based Intrusion Detection System) se concentre principalement sur les comportements des terminaux en surveillant l'activité des systèmes d'exploitation, des fichiers journaux et des applications installées sur les machines individuelles, tandis que le NIDS (Network-based Intrusion Detection System) est chargé de surveiller le trafic réseau en temps réel afin de détecter des anomalies ou des schémas d'attaque caractéristiques circulant sur le réseau [42]. Le NIDS présente l'avantage de pouvoir détecter certaines attaques dès leur phase initiale, en identifiant des flux malveillants avant même qu'ils n'atteignent leur cible, tandis que le HIDS intervient généralement en phase postérieure, en détectant les signes d'une compromission une fois que l'attaque a potentiellement déjà atteint le système hôte. Pour assurer une protection optimale et globale des infrastructures informatiques, il est recommandé de déployer simultanément ces deux types de systèmes, car ils se complètent en couvrant à la fois les aspects réseau et système local. Les solutions modernes de gestion des informations et des événements de sécurité (SIEM Security Information and Event Management) intègrent les données collectées par les NIDS et les HIDS, permettant ainsi une corrélation en temps réel des événements de sécurité et une détection plus efficace des incidents complexes. Le choix du type d'IDS le plus approprié dépend toutefois des besoins spécifiques de chaque organisation, en fonction de la nature de ses ressources critiques, de son architecture réseau et des menaces auxquelles elle est exposée.

1.6.3 IDS hybride (HIDS + NIDS)

Le système de détection d'intrusion hybride (Hybrid IDS) désigne l'intégration simultanée des fonctionnalités du HIDS (Host-based Intrusion Detection System) et du NIDS (Network-based Intrusion Detection System) afin de constituer une solution de sécurité plus robuste, polyvalente et complète face aux menaces modernes [43]. Cette approche hybride se révèle particulièrement efficace dans les environnements complexes où de multiples points d'entrée peuvent être exploités par les attaquants, ou lorsque les défenses périmétriques traditionnelles ne suffisent plus à garantir la sécurité du réseau. Les systèmes hybrides sont également capables de détecter des menaces avancées et furtives, telles que les attaques persistantes avancées (APT), qui sont souvent conçues pour contourner les mécanismes de défense classiques. Parmi les solutions hybrides les plus utilisées figurent Snort, OSSEC et Suricata, qui associent des méthodes de détection basées sur les signatures (en s'appuyant sur des bases de données de menaces connues) et sur les anomalies (en identifiant les comportements inhabituels), souvent renforcées par l'intégration de techniques d'apprentissage automatique (machine learning) pour améliorer la détection des comportements malveillants émergents. Grâce à cette approche combinée, les systèmes IDS hybrides permettent une couverture beaucoup plus large et une détection plus fine des incidents de sécurité affectant à la fois les hôtes et l'infrastructure réseau. La Figure 1.11 illustre le fonctionnement d'un système de détection d'intrusion hybride.

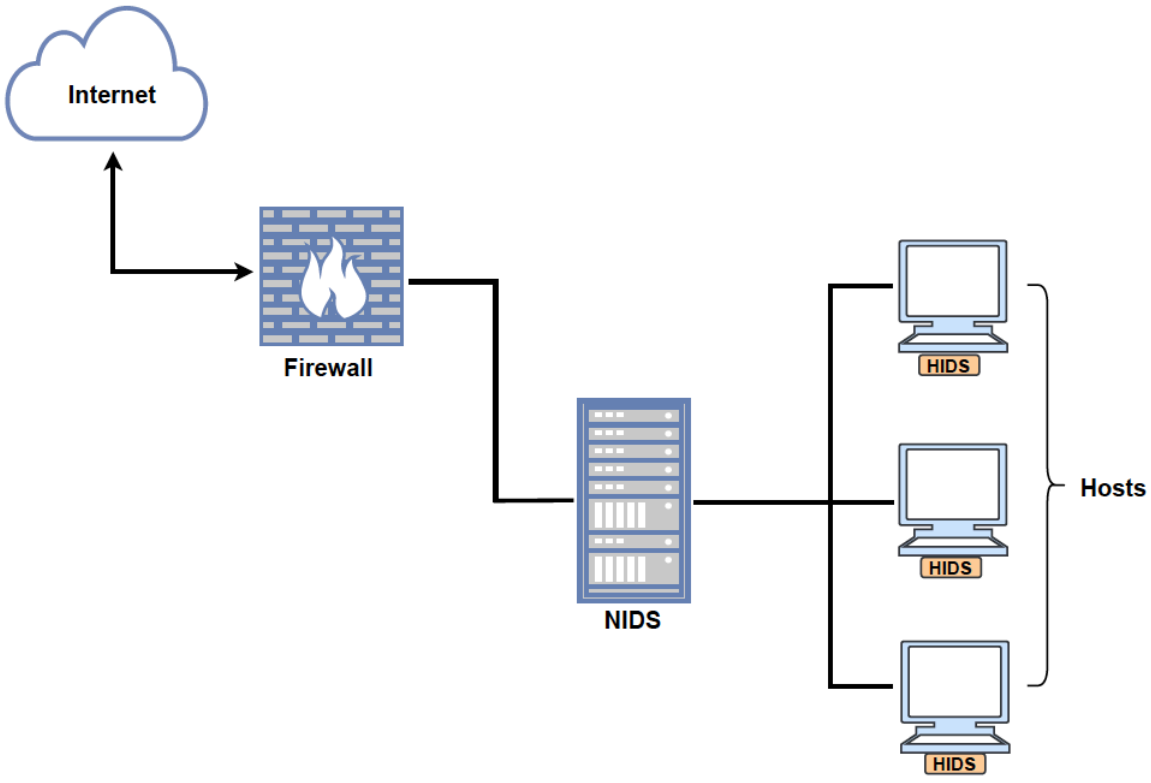


Figure 1.11 *– IDS hybride.*

1.7 Méthodes de détection IDS

Les systèmes de détection d'intrusion utilisent principalement deux techniques pour détecter les menaces et alerter les administrateurs de réseau : les techniques basées sur les signatures et les techniques basées sur les anomalies [44].

1.7.1 Détection basée sur la signature

La détection basée sur les signatures (signature-based detection) constitue l'une des méthodes les plus couramment utilisées dans les systèmes de détection d'intrusion [43]. Son fonctionnement repose sur la comparaison du trafic réseau entrant avec une base de données contenant des empreintes numériques ou signatures d'attaques connues, permettant ainsi d'identifier rapidement les comportements malveillants lorsqu'une correspondance est trouvée. Cette approche est particulièrement efficace pour détecter les attaques largement répertoriées et fréquentes telles que les infections par malwares, les campagnes de phishing ou les attaques par déni de service (DoS), et présente l'avantage d'être relativement simple à déployer et à gérer, puisque les bases de signatures peuvent être mises à jour régulièrement par les éditeurs de sécurité ou les experts en cybersécurité. Toutefois, la détection par signatures présente certaines limitations. Elle est incapable d'identifier les attaques nouvelles, inconnues (zero-day), ou les variantes d'attaques existantes qui ne correspondent pas aux signatures répertoriées dans la base de données. De plus, elle peut générer des faux positifs, en identifiant à tort des activités légitimes comme étant malveillantes, ce qui peut perturber l'analyse de sécurité. Par ailleurs, cette méthode peut s'avérer gourmande en ressources, car chaque paquet de données doit être comparé à un grand nombre de signatures, ce qui, dans certains cas, peut ralentir les performances globales du réseau.

1.7.2 Détection basée sur les anomalies

La détection basée sur les anomalies (anomaly-based detection) représente une approche moderne et avancée des systèmes de détection et de prévention des intrusions (IDPS), s'appuyant sur l'analyse statistique, l'apprentissage automatique (machine learning) et l'intelligence artificielle pour établir un modèle de comportement normal du réseau [43]. Une fois ce modèle de référence construit, le système surveille le trafic réseau à la recherche d'activités inhabituelles ou déviantes susceptibles d'indiquer une attaque potentielle. Lorsqu'une anomalie est détectée, l'IDPS peut soit générer une alerte à destination des administrateurs de sécurité, soit bloquer automatiquement le trafic suspect. Cette méthode est particulièrement efficace pour identifier des attaques nouvelles ou inconnues (zero-day) ainsi que des variantes d'attaques existantes qui n'ont pas encore de signature répertoriée, offrant ainsi une protection proactive

contre des menaces émergentes. De plus, la détection basée sur les anomalies est dynamique et évolutive : elle est capable d'adapter en continu son modèle de normalité en fonction des évolutions du comportement du réseau, améliorant ainsi sa capacité de détection au fil du temps. Toutefois, cette approche présente également certaines limites. L'établissement et la maintenance d'une ligne de base fiable pour caractériser le comportement normal peuvent s'avérer complexes, en particulier dans les réseaux hétérogènes ou en constante évolution. Par ailleurs, des faux négatifs peuvent survenir lorsque des attaques sont habilement déguisées en trafic légitime ou imitent des comportements habituels. Enfin, la mise en uvre et l'exploitation de solutions de détection basées sur les anomalies nécessitent des ressources technologiques importantes ainsi qu'une expertise avancée, ce qui peut engendrer des coûts d'installation et de maintenance relativement élevés.

1.8 Architecture des système de détection d'intrusion

L'architecture d'un IDS peut varier en fonction de la mise en uvre et des exigences spécifiques du système [45], mais les trois principaux composants de l'architecture d'un IDS sont les suivants :

1.8.1 Collecte de données

Les capteurs jouent un rôle fondamental dans les systèmes de détection d'intrusion (IDS) en assurant la collecte de données issues de multiples sources, telles que le trafic réseau, les journaux système (logs), ainsi que d'autres sources d'événements, permettant ainsi d'identifier d'éventuelles menaces de sécurité [43]. Les IDS s'appuient sur ces capteurs pour obtenir des informations précieuses provenant de différents environnements : les systèmes de détection d'intrusion réseau (Network Intrusion Detection Systems NIDS) surveillent le trafic réseau en temps réel ; les systèmes de détection d'intrusion basés sur l'hôte (Host-based Intrusion Detection Systems HIDS) analysent l'activité au niveau des systèmes individuels ; et les systèmes de détection d'intrusion applicative (Application-based Intrusion Detection Systems AIDS) se concentrent sur les applications spécifiques. Ces capteurs spécialisés permettent d'observer en profondeur les différentes couches d'un environnement informatique afin de détecter des comportements anormaux ou malveillants. Parmi les technologies de capteurs les plus couramment utilisées dans les déploiements IDS figurent Snort, connu pour son efficacité en détection basée sur les signatures, Suricata, qui combine analyse protocolaire avancée et inspection des flux, Zeek (anciennement Bro), reconnu pour sa capacité à effectuer des analyses comportementales approfondies, et OSSEC, un HIDS open source largement utilisé pour la surveillance des journaux et des intégrités

de fichiers au niveau des systèmes d'exploitation. Grâce à la diversité de ces capteurs, les systèmes IDS bénéficient d'une couverture étendue et d'une capacité de détection optimisée face à la complexité croissante des cybermenaces contemporaines.

1.8.2 Prétraitement des données

Le prétraitement des données (data pre-processing) constitue une étape essentielle dans le processus d'analyse de sécurité, consistant à filtrer, normaliser et extraire les caractéristiques pertinentes des données collectées afin de générer des enregistrements d'activité exploitables pour l'analyse des incidents et la détection des menaces [43]. Cette phase permet de transformer les volumes massifs de données brutes en ensembles d'informations structurées et ciblées, facilitant ainsi leur traitement ultérieur par les systèmes de détection. Les données prétraitées peuvent inclure des éléments tels que les adresses IP source et destination, les numéros de port, les types de protocoles utilisés, les horodatages des communications ainsi que les charges utiles des paquets (packet payloads). L'objectif principal du prétraitement est de réduire la quantité de données à analyser, tout en améliorant leur qualité et leur cohérence, de manière à rendre les processus d'analyse plus efficaces et plus précis. Ce filtrage initial permet également de minimiser le bruit dans les données, d'éliminer les redondances et de mieux isoler les indicateurs de compromission susceptibles de révéler des comportements malveillants ou anormaux. Un prétraitement rigoureux constitue ainsi un prérequis indispensable pour garantir la performance et la fiabilité des systèmes d'analyse comportementale, des IDS/IPS et des solutions d'intelligence artificielle appliquées à la cybersécurité.

1.8.3 Reconnaissance des intrusions

La reconnaissance d'intrusion (intrusion recognition) désigne le processus d'identification et de classification des menaces de sécurité détectées par un système de détection d'intrusion (IDS), en s'appuyant sur différentes techniques telles que la détection basée sur les signatures, la détection basée sur les anomalies ou encore des méthodes reposant sur l'apprentissage automatique (Machine Learning) [43]. Chaque approche permet d'analyser les comportements du réseau et du système afin d'identifier la présence d'activités malveillantes et de les catégoriser en fonction de leur nature et de leur gravité. L'objectif fondamental de l'architecture IDS est de détecter rapidement et avec précision les menaces de sécurité afin d'en limiter l'impact sur les systèmes ou les réseaux protégés. Pour atteindre cet objectif, l'IDS repose sur une interaction étroite entre plusieurs modules complémentaires : la collecte de données, qui agrège les informations issues de multiples sources ; le prétraitement des données, qui filtre, normalise et extrait les caractéristiques significatives ; et la reconnaissance d'intrusion, qui analyse ces données traitées afin de détecter et classer les attaques.

Cette synergie fonctionnelle entre les différentes composantes de l'IDS est représentée visuellement dans la Figure 1.12, illustrant ainsi l'enchaînement logique des étapes essentielles à une détection d'intrusion efficace.

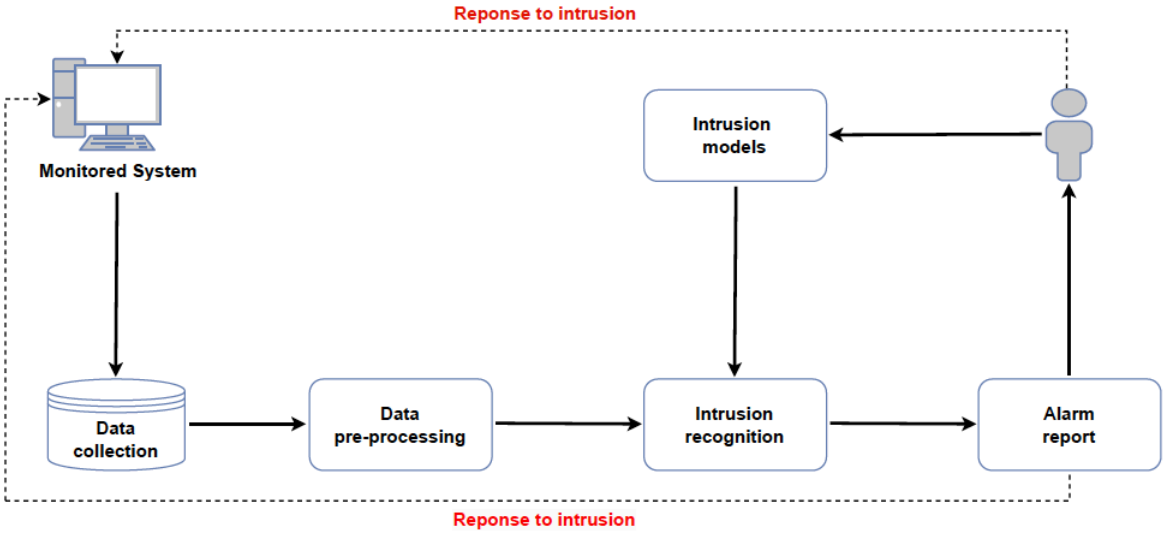


Figure 1.12 *– Cadre du système de détection d'intrusion.*

1.9 Scénarios de déploiement des Systèmes de détection d'intrusion

Les scénarios de déploiement des IDS [3] font référence aux différentes manières dont les systèmes de détection d'intrusion sont positionnés et configurés au sein d'un réseau. Ces scénarios peuvent être classés en trois grandes catégories :

1.9.1 IDS basés sur le périmètre

Le système de détection d'intrusion périmétrique (Perimeter-based IDS) est déployé à la périphérie du réseau, généralement à l'interface entre le réseau interne et Internet. Sa fonction principale consiste à surveiller et filtrer le trafic entrant provenant de l'extérieur afin de détecter les attaques externes telles que les scans de ports, les tentatives de reconnaissance de vulnérabilités ou les infections par logiciels malveillants. Le système IDS périmétrique est généralement mis en uvre sous la forme d'un système de détection d'intrusion basé sur le réseau (Network-based IDS NIDS), à l'aide de capteurs ou de sondes qui capturent et analysent le trafic réseau en temps réel. Il constitue souvent la première ligne de défense contre les menaces externes et vient en complément d'autres dispositifs de sécurité tels que les pare-feu et les systèmes de prévention d'intrusion (Intrusion Prevention Systems IPS).

1.9.2 IDS interne

Le système de détection d'intrusion interne (Internal-based IDS) est déployé à l'intérieur du réseau interne, généralement aux niveaux des points ou segments critiques où sont localisées des données sensibles ou des systèmes stratégiques. Sa fonction principale est de détecter et de répondre aux menaces internes, telles que les attaques provenant d'utilisateurs internes malveillants, les mouvements latéraux des attaquants dans le réseau, ou les accès non autorisés aux ressources. L'IDS interne peut être implémenté sous la forme d'un système de détection d'intrusion basé sur l'hôte (Host-based IDS HIDS), en utilisant des agents ou des logiciels qui surveillent et analysent le comportement des hôtes ou des terminaux individuels, ou sous la forme d'un système basé sur le réseau (Network-based IDS NIDS), en recourant à des capteurs ou des sondes qui surveillent et analysent le trafic réseau interne. Le déploiement d'un IDS interne constitue un complément précieux aux contrôles d'accès, aux systèmes de gestion des identités et aux solutions de prévention des pertes de données (Data Loss Prevention DLP).

1.9.3 IDS distribué

Le système de détection d'intrusion distribué (Distributed IDS) constitue un scénario de déploiement hybride combinant à la fois les approches périmétrique et interne. Il consiste à déployer plusieurs capteurs ou sondes à des emplacements stratégiques à travers l'ensemble du réseau, aussi bien à sa périphérie qu'en son cur, afin d'assurer une couverture et une visibilité globales sur l'ensemble des activités du système d'information. Le Distributed IDS offre ainsi une solution flexible et évolutive, capable de s'adapter à différentes architectures réseau, topologies et exigences de sécurité. De plus, ce type de déploiement permet d'assurer des fonctions de redondance, d'équilibrage de charge et de reprise automatique en cas de défaillance de capteurs ou de pannes réseau. Le choix du scénario de déploiement IDS approprié constitue une étape critique, chaque approche présentant ses propres avantages et défis, qui varient en fonction de l'environnement réseau, du contexte des menaces et des objectifs de sécurité poursuivis. Il est par conséquent essentiel d'évaluer soigneusement les besoins spécifiques et les contraintes propres à chaque organisation afin de sélectionner le modèle de déploiement IDS le mieux adapté.

1.10 Conclusion

En conclusion, la sécurité des réseaux constitue un aspect fondamental de l'infrastructure informatique de toute organisation, d'autant plus que le paysage des menaces ne cesse d'évoluer. Comme cela a été présenté dans ce chapitre, les réseaux sont exposés à diverses menaces telles que les attaques par déni de service (DoS),

déni de service distribué (DDoS), les attaques par force brute, les injections SQL, les infiltrations et les réseaux de zombies (botnets). Pour se prémunir contre ces menaces, les organisations peuvent déployer plusieurs mesures de sécurité, notamment les pare-feu, les réseaux privés virtuels (VPN), le chiffrement des données, le contrôle d'accès, les systèmes de gestion des informations et des événements de sécurité (SIEM), ainsi que les systèmes de détection et de prévention des intrusions (IDPS). Les systèmes de détection d'intrusion constituent une composante essentielle de la sécurité des réseaux, avec différentes variantes telles que les systèmes basés sur le réseau (NIDS), basés sur l'hôte (HIDS) et les systèmes hybrides. Le choix du scénario de déploiement d'un IDS qu'il soit périmétrique, interne ou distribué doit être effectué en fonction des besoins spécifiques et des contraintes propres à chaque organisation. De manière générale, les organisations doivent rester vigilantes et adopter une approche proactive en matière de cybersécurité afin de réduire les risques et de prévenir les cyberattaques potentielles. Dans le chapitre suivant, nous introduirons les concepts d'apprentissage profond (Deep Learning) et d'apprentissage automatique (Machine Learning).

Chapitre 2

Intelligence artificielle, Machine Learning et Deep Learning

Sommaire

2.1 Introduction

L'objectif principal de l'intelligence artificielle (IA) est de doter les ordinateurs de la capacité de comprendre et d'interagir intelligemment avec le monde réel. L'apprentissage profond (Deep Learning, DL) s'est imposé comme une approche de pointe pour réaliser cette ambition. Il repose sur l'utilisation de réseaux neuronaux profonds hiérarchiques, capables d'extraire automatiquement des représentations de haut niveau à partir de grandes quantités de données, ce qui le rend particulièrement efficace dans des domaines complexes tels que la vision par ordinateur, le traitement du langage naturel et la reconnaissance vocale [46, 47]. L'apprentissage profond constitue une branche spécifique de l'apprentissage automatique (Machine Learning), d'où la nécessité de maîtriser les fondements de ce dernier pour comprendre pleinement les mécanismes sous-jacents du DL. En effet, plusieurs concepts centraux des réseaux neuronaux comme la rétropropagation, les fonctions de perte ou les algorithmes d'optimisation trouvent leur origine dans les théories plus générales de l'apprentissage automatique [48]. La révision de ces notions facilite ainsi la compréhension et l'application des méthodes spécifiques à l'apprentissage profond.

L'apprentissage automatique et l'apprentissage profond sont aujourd'hui largement déployés dans une grande diversité de secteurs. En médecine, ces approches permettent d'améliorer les diagnostics précoces, de soutenir la médecine personnalisée et d'automatiser l'analyse d'images médicales [49, 50]. En finance, elles sont utilisées pour la détection de fraudes, la gestion automatisée des portefeuilles et l'analyse prédictive des marchés [51]. Dans le commerce électronique, elles soutiennent les systèmes de recommandation personnalisée, l'optimisation de l'expérience utilisateur et la segmentation des clients [52]. Dans les transports, elles servent à la prévision du trafic, à la maintenance prédictive, ainsi qu'au développement des véhicules autonomes [53].

En traitement du langage naturel, elles permettent la création d'assistants virtuels, de systèmes de traduction automatique et de chatbots intelligents [54]. Enfin, en reconnaissance d'images et de vidéos, elles sont employées pour la détection d'objets, la reconnaissance faciale et la surveillance automatisée [55]. Ces exemples illustrent la portée croissante des applications de l'intelligence artificielle, portée en grande partie par les avancées en apprentissage automatique et profond.

L'apprentissage profond constitue un sous-ensemble de l'apprentissage automatique, lequel relève lui-même du domaine plus vaste de l'intelligence artificielle (comme illustré dans la Figure 2.1).

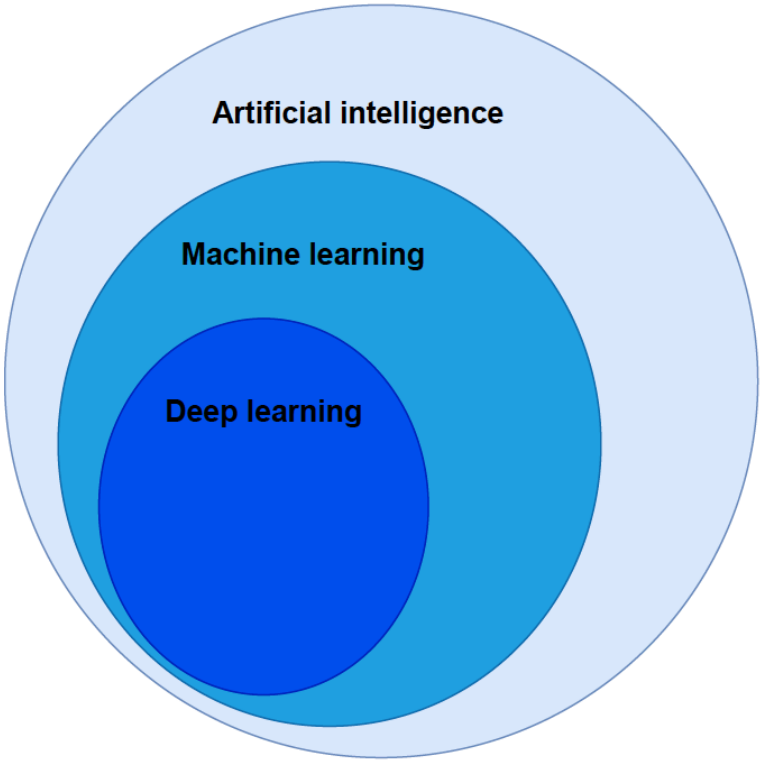


Figure 2.1 *– Représentation de la relation entre IA, ML, et DL dans le diagramme de Venn.*

2.2 Principes fondamentaux de l'apprentissage automatique

L'apprentissage automatique (Machine Learning) fait référence à la capacité des ordinateurs à apprendre automatiquement des modèles et des relations à partir des données, sans qu'ils soient explicitement programmés à cet effet. Il repose sur l'utilisation de modèles et de techniques statistiques permettant à l'ordinateur d'extraire des schémas à partir des données, puis d'appliquer ces connaissances afin de réaliser des prédictions ou de prendre des décisions, et ce, sans nécessiter de programmation

explicite. Cette capacité d'apprentissage et d'adaptation s'appuie généralement sur l'exploitation de grandes quantités de données.

L'apprentissage automatique constitue un domaine diversifié et en constante évolution, qui mobilise une variété de méthodes pour atteindre ses objectifs. Parmi les approches les plus courantes figurent l'apprentissage par renforcement, l'apprentissage supervisé et l'apprentissage non supervisé.

- **Apprentissage par renforcement :** L'apprentissage par renforcement constitue une approche dans laquelle l'ordinateur acquiert progressivement des connaissances par un processus d'essais et d'erreurs, en interagissant de manière continue avec un environnement donné et en recevant des rétroactions sous forme de récompenses ou de pénalités. À chaque interaction, l'agent prend une décision, observe le résultat de cette action, et ajuste son comportement en fonction de la qualité de la récompense obtenue. L'objectif fondamental de l'apprentissage par renforcement est de découvrir une politique optimale, c'est-à-dire une séquence d'actions ou une stratégie décisionnelle qui permet de maximiser la récompense cumulée au fil du temps. Cette approche trouve de nombreuses applications dans des domaines variés tels que le contrôle robotique, les jeux, la gestion des ressources ou encore la prise de décision autonome [56].
- **Apprentissage non supervisé :** L'apprentissage non supervisé constitue une approche dans laquelle l'ordinateur reçoit un ensemble de données non étiquetées, c'est-à-dire dépourvues de variables de sortie ou de cibles prédéfinies. Contrairement à l'apprentissage supervisé, aucune correspondance explicite entre les entrées et les sorties n'est fournie au système. L'objectif principal de l'apprentissage non supervisé est de permettre à l'algorithme de découvrir de manière autonome des structures sous-jacentes, des motifs récurrents ou des relations latentes au sein des données. Parmi les tâches courantes associées à cette approche figurent le regroupement d'exemples similaires en classes homogènes (clustering) et l'identification d'observations atypiques ou aberrantes (détection d'outliers). Ces méthodes sont largement utilisées dans des domaines tels que l'analyse exploratoire de données, la réduction de dimensionnalité, la segmentation de clientèle, ou encore la détection de fraudes [57].
- **Apprentissage supervisé :** L'apprentissage supervisé repose sur l'utilisation d'un ensemble de données étiquetées, dans lequel chaque exemple est associé à une sortie ou à une variable cible correcte. L'objectif de cette approche est de permettre à l'algorithme d'apprendre une fonction de correspondance entre les entrées et les sorties, de façon à pouvoir effectuer des prédictions précises lorsqu'il est confronté à de nouvelles données non vues au préalable. Durant la phase d'entraînement, le modèle ajuste ses paramètres internes en minimisant l'écart entre les prédictions générées et les résultats attendus, souvent à l'aide de fonctions de coût et de techniques d'optimisation numérique. L'apprentis-

sage supervisé est particulièrement adapté aux problèmes de classification, où l'on cherche à assigner des catégories aux observations, ainsi qu'aux tâches de régression, où l'on prédit des valeurs continues. Cette méthode est largement appliquée dans de nombreux domaines, tels que la reconnaissance d'images, la détection de spams, la prévision financière ou encore le diagnostic médical [58].

- **Apprentissage semi-supervisé :** L'apprentissage semi-supervisé constitue une approche hybride située à mi-chemin entre l'apprentissage supervisé et l'apprentissage non supervisé. Dans ce paradigme, l'algorithme est entraîné à partir d'un ensemble de données comprenant à la fois des exemples étiquetés (disposant de variables cibles connues) et un grand volume de données non étiquetées. L'objectif est d'exploiter efficacement l'information contenue dans les données non étiquetées afin d'améliorer la performance du modèle, tout en minimisant le besoin coûteux et chronophage de labellisation manuelle de grandes quantités de données. Cette approche est particulièrement utile lorsque l'acquisition des étiquettes est complexe, onéreuse ou dépend de l'expertise humaine, comme c'est souvent le cas en médecine, en vision par ordinateur ou en traitement du langage naturel. Les techniques d'apprentissage semi-supervisé incluent, entre autres, les méthodes de propagation de labels, les modèles génératifs, les autoencodeurs, ainsi que les approches basées sur la régularisation de la consistance entre les prédictions [59].

Parmi les trois approches, l'apprentissage supervisé est souvent considéré comme le plus intéressant, car il s'apparente étroitement au mode d'apprentissage des êtres humains. En effet, tout comme un enseignant fournit des exemples et des corrections à un élève, l'apprentissage supervisé permet au modèle d'apprendre à partir d'exemples annotés, en ajustant progressivement ses prédictions en fonction des erreurs observées.

2.3 Apprentissage supervisé

L'apprentissage supervisé constitue un outil puissant pour résoudre un large éventail de problèmes, tels que la reconnaissance d'images, le traitement du langage naturel ou encore les systèmes de recommandation [57]. L'un des avantages majeurs de l'apprentissage supervisé réside dans sa capacité à construire des modèles complexes capables de capturer des relations non linéaires entre les variables d'entrée et les variables de sortie, ce qui permet d'obtenir des prédictions précises même dans des contextes fortement complexes et multidimensionnels (voir Figure 2.2).

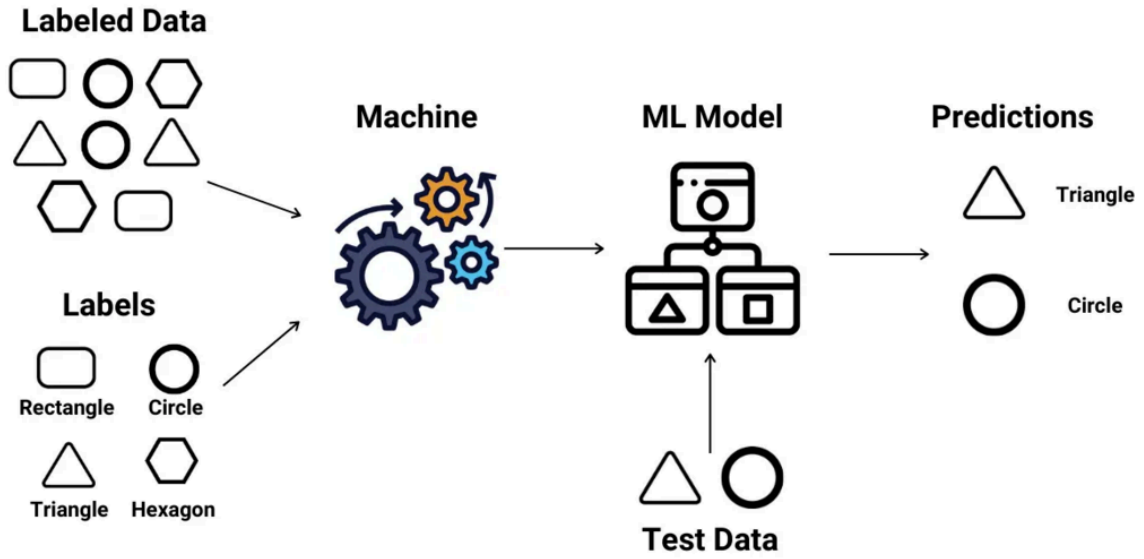


Figure 2.2 *– Apprentissage automatique supervisé.*

Pour maîtriser l'apprentissage supervisé, il est essentiel de comprendre et de maîtriser les quatre concepts fondamentaux suivants :

- Dataset.
- Le modèle et ses paramètres.
- Fonction de coût.
- Algorithme d'apprentissage.

2.3.1 Dataset

Le dataset constitue un élément central dans le cadre de l'apprentissage supervisé. Il s'agit d'un ensemble structuré de paires entrée/sortie, utilisé à la fois pour l'entraînement et l'évaluation du modèle. Les données d'entrée représentent les caractéristiques ou attributs pertinents du problème étudié, souvent appelés features, tandis que les données de sortie correspondent aux cibles ou aux étiquettes (labels) que le modèle cherche à prédire. La qualité, la représentativité et la diversité du jeu de données influencent directement la capacité du modèle à généraliser et à produire des prédictions fiables sur de nouvelles données non vues. Un jeu de données bien équilibré et soigneusement préparé est donc fondamental pour assurer la robustesse et la performance des modèles d'apprentissage supervisé.

2.3.2 Le modèle et ses paramètres

Le modèle constitue la représentation mathématique de la relation existant entre les données d'entrée et les données de sortie. Il est défini par un ensemble de paramètres internes qui sont ajustés au cours du processus d'apprentissage, dans le but de minimiser l'écart entre les sorties prédites par le modèle et les sorties réelles observées

dans le jeu de données d'entraînement. Ce processus d'ajustement repose généralement sur des techniques d'optimisation numérique visant à réduire une fonction de coût ou une fonction de perte mesurant l'erreur de prédiction [46]. Le choix du type de modèle (par exemple : régression linéaire, réseaux neuronaux, machines à vecteurs de support) ainsi que la sélection appropriée de ses hyperparamètres jouent un rôle déterminant sur la qualité des prédictions et sur la capacité du modèle à généraliser sur de nouvelles données [48]. Une modélisation adaptée permet ainsi de capturer les relations sous-jacentes, qu'elles soient linéaires ou non linéaires, complexes ou simples, entre les variables étudiées.

2.3.3 Fonction de coût

La fonction de coût, également appelée fonction de perte, est une fonction mathématique qui quantifie l'écart entre les sorties prédites par le modèle et les sorties réelles observées pour un ensemble donné de paramètres du modèle. Elle constitue un indicateur essentiel permettant d'évaluer la qualité des prédictions effectuées par le modèle au cours de l'apprentissage. L'objectif de l'algorithme d'apprentissage est ainsi de rechercher les valeurs optimales des paramètres du modèle qui minimisent cette fonction de coût, en réduisant progressivement l'erreur de prédiction sur les données d'entraînement [60]. Différents types de fonctions de coût peuvent être utilisés en fonction de la nature du problème traité, qu'il s'agisse de tâches de classification ou de régression, telles que l'erreur quadratique moyenne (mean squared error) ou l'entropie croisée (cross-entropy). Ce processus d'optimisation est représenté de manière schématique à la Figure 2.3.

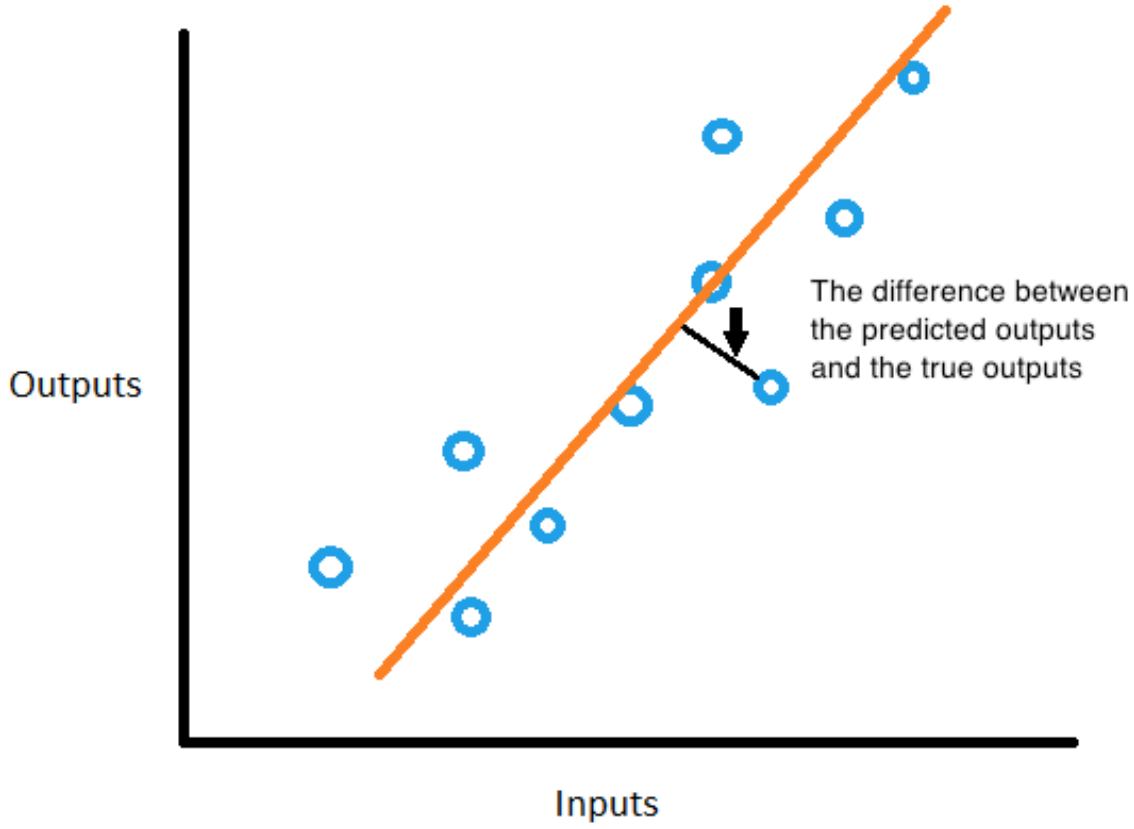


Figure 2.3 *– Représentation de fonction coût.*

2.3.4 Algorithme d'apprentissage

Un algorithme d'apprentissage est un ensemble de règles et de procédures permettant de mettre à jour les paramètres du modèle en fonction de l'écart observé entre les sorties prédites et les sorties réelles. Le choix de l'algorithme d'apprentissage peut avoir un impact considérable sur les performances du modèle, et il existe une grande variété d'algorithmes adaptés aux différentes natures de problèmes rencontrés [61,62].

2.4 Avantages de l'apprentissage profond par rapport aux algorithmes traditionnels d'apprentissage automatique

Les algorithmes d'apprentissage profond (Deep Learning) apprennent automatiquement à extraire des caractéristiques de haut niveau à partir de données brutes, en s'appuyant sur des réseaux neuronaux artificiels composés de multiples couches successives. Cette architecture hiérarchique permet aux modèles d'apprentissage profond d'exploiter de très grandes quantités de données et d'accomplir des tâches complexes, telles que la reconnaissance d'images ou le traitement du langage naturel, avec un ni-

veau de précision élevé [63–65]. En revanche, les algorithmes d'apprentissage automatique traditionnels (Machine Learning) reposent sur l'utilisation de caractéristiques extraites manuellement et sur des méthodes statistiques pour analyser les données et effectuer des prédictions. Dans ces approches classiques, l'intervention d'un expert humain est souvent nécessaire pour concevoir et sélectionner les variables pertinentes qui seront ensuite introduites dans le modèle. Les algorithmes d'apprentissage automatique permettent de réaliser des tâches telles que la classification ou la régression ; cependant, ils se révèlent généralement moins performants que les algorithmes d'apprentissage profond lorsqu'il s'agit de traiter des données complexes, volumineuses et non structurées (voir Figure 2.4).

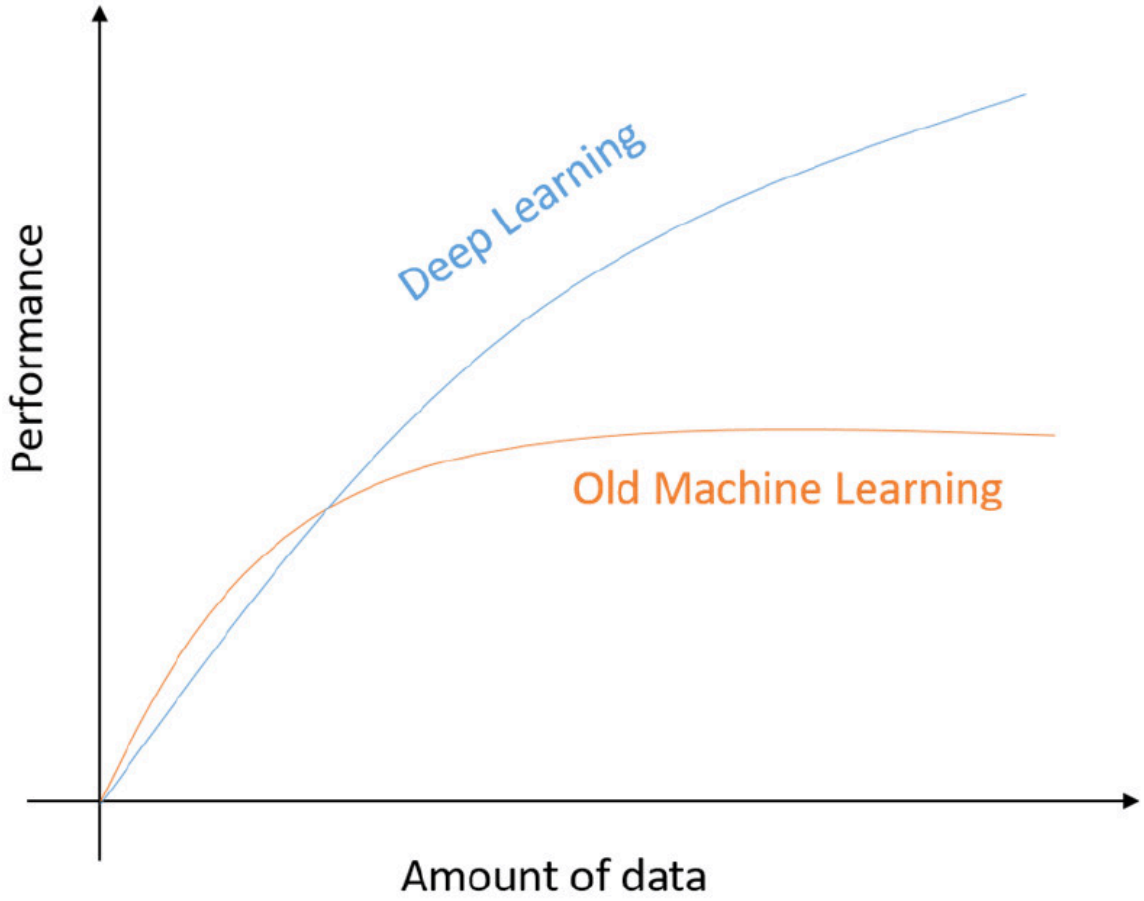


Figure 2.4 *– L'impact de la disponibilité des données sur la performance des algorithmes.*

2.5 Introduction au Deep Learning

L'apprentissage profond constitue une branche avancée de l'apprentissage automatique, reposant sur l'utilisation de réseaux neuronaux comportant de multiples couches, afin de faciliter et d'améliorer le processus d'apprentissage. Cette approche est particulièrement adaptée au traitement de données massives et complexes, qu'il

s'agisse d'images, de textes ou de signaux audio, en permettant aux modèles d'extraire automatiquement des représentations hiérarchiques des données à partir d'informations brutes, sans nécessiter de prétraitement ou d'extraction manuelle de caractéristiques par un expert. L'apprentissage profond s'appuie sur le fonctionnement des réseaux neuronaux artificiels, dont l'architecture s'inspire de la structure et du mode de fonctionnement du cerveau humain, notamment par la présence de neurones interconnectés et de couches de traitement successives permettant des transformations progressives de l'information [66, 67].

2.6 Réseaux de neurones artificiels

Les réseaux neuronaux artificiels simulent le mécanisme d'apprentissage observé chez les organismes biologiques. Dans le système nerveux humain, l'unité fonctionnelle de base est constituée par les cellules nerveuses, appelées neurones. Chaque neurone est constitué d'un corps cellulaire, de prolongements appelés dendrites qui reçoivent les signaux provenant d'autres neurones, et d'un axone qui transmet les signaux vers d'autres cellules. Les points de connexion entre les axones et les dendrites d'autres neurones sont désignés sous le terme de synapses [68]. C'est au niveau de ces synapses que s'effectue la transmission de l'information ainsi que la modulation des signaux en fonction de l'intensité et de la fréquence des stimulations. L'architecture des réseaux neuronaux artificiels s'inspire de cette organisation biologique, en représentant les neurones artificiels par des unités interconnectées capables de traiter et de transmettre des informations selon des poids synaptiques ajustables. Cette structure est représentée de manière schématique à la Figure 2.5.

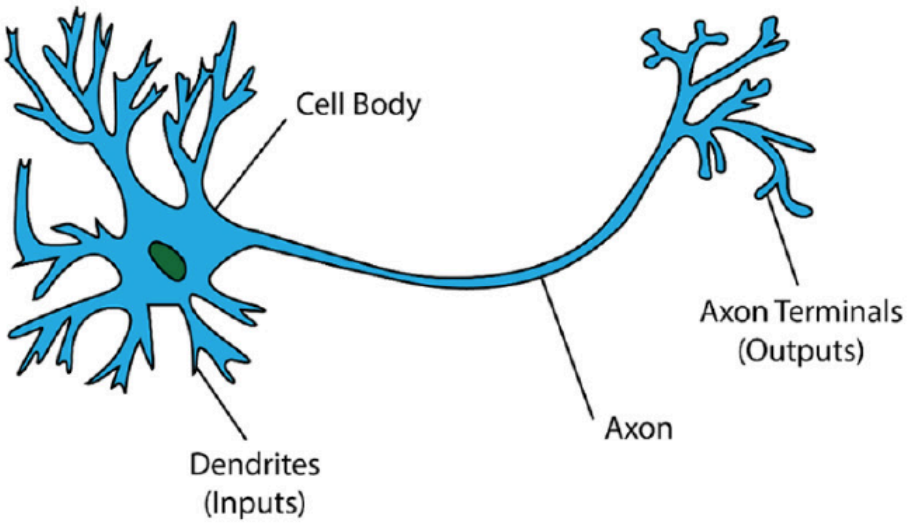


Figure 2.5 *– Structure d'un neurone biologique.*

Les dendrites reçoivent les signaux d'entrée provenant d'autres neurones par l'in-

termédiaire des synapses. Ces signaux, sous forme de potentiels post-synaptiques, sont ensuite intégrés au niveau du corps cellulaire (soma). Lorsque la somme des signaux excède un certain seuil d'excitation, le neurone génère un potentiel d'action qui est propagé le long de l'axone. Ce signal électrique, une fois arrivé aux terminaisons axonales, est transmis aux neurones suivants par l'intermédiaire de nouvelles synapses, assurant ainsi la communication et la transmission de l'information au sein du système nerveux.

2.6.1 Composants d'un réseau de neurones artificiel

Les réseaux neuronaux artificiels (ANN, Artificial Neural Networks) sont constitués de plusieurs composants qui collaborent afin de traiter l'information et de prendre des décisions. Parmi ces éléments, on retrouve les neurones (également appelés perceptrons), les poids synaptiques, les fonctions d'activation et les couches. Les neurones représentent les unités de traitement fondamentales des ANN, recevant et transformant les signaux d'entrée. Les poids, quant à eux, modulent l'intensité des connexions entre les neurones, en reflétant l'importance relative de chaque entrée. Les fonctions d'activation permettent de déterminer si un neurone doit s'activer ou non en fonction de la somme pondérée de ses entrées, introduisant ainsi des comportements non linéaires indispensables à la modélisation de phénomènes complexes. Les couches, organisées de manière hiérarchique (couches d'entrée, cachées et de sortie), réalisent des calculs intermédiaires qui permettent l'extraction progressive de caractéristiques à différents niveaux d'abstraction. Une bonne compréhension de ces différents composants et de leurs interactions est essentielle pour appréhender le fonctionnement des réseaux neuronaux artificiels et pour exploiter leur potentiel dans la résolution de problèmes complexes [69–71].

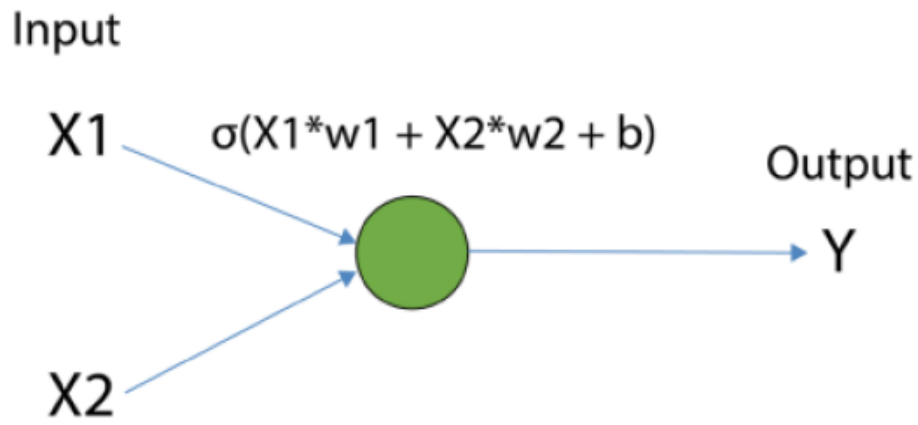


Figure 2.6 *– Représentation d'un réseau de neurones avec deux entrées.*

Dans les modèles d'apprentissage profond, les paramètres sont désignés sous les termes de poids (w) et de biais (b). Ces paramètres interviennent dans le calcul de

la somme pondérée des entrées, en multipliant chaque entrée par son poids respectif, puis en ajoutant un terme de biais au résultat obtenu. Ce mécanisme est illustré dans le calcul effectué au niveau du nud représenté dans l'image précédente. Plus précisément, les entrées sont notées X1 et X2, les poids correspondants sont w1 et w2, et le biais est b. La somme des produits pondérés et du biais constitue ensuite l'entrée d'une fonction non linéaire appelée fonction d'activation (σ), comme le montre la Figure 2.6. Le résultat de cette fonction correspond à l'activation du neurone, c'est-à-dire la sortie produite par celui-ci après traitement de ses entrées, en tenant compte des poids, du biais et de la fonction d'activation choisie [65].

2.6.2 Réseaux neuronaux progressifs (FFNNs)

Le réseau neuronal à propagation avant (Feedforward Neural Network, FFNN) est un type de réseau neuronal artificiel dans lequel le traitement de l'information s'effectue de manière unidirectionnelle, en progressant des neurones d'entrée vers les neurones de sortie, sans rétroaction ni boucle de rétropropagation interne [72]. Ce type d'architecture se compose d'une couche d'entrée, d'une ou de plusieurs couches cachées, ainsi que d'une couche de sortie. Au cours de la phase d'apprentissage, les poids et les biais associés aux connexions neuronales sont ajustés itérativement dans le but de minimiser l'écart entre la sortie prédite par le réseau et la sortie réelle attendue. Cette optimisation est généralement réalisée en appliquant des algorithmes de rétropropagation de l'erreur combinés à des techniques de descente de gradient, permettant ainsi au réseau d'améliorer progressivement ses performances prédictives.

2.6.2.1 Architecture du FFNNs

Pour construire un réseau neuronal monocouche, les neurones sont empilés les uns à côté des autres au sein d'une même couche, comme illustré à la Figure 2.7. Dans cette architecture, chaque valeur d'entrée provenant de la couche d'entrée est transmise à l'ensemble des neurones de la couche cachée, créant ainsi une connexion complète entre les deux couches. Pour chaque neurone de la couche cachée, le calcul de la somme pondérée des entrées et du biais est effectué de manière indépendante. Cette somme est ensuite soumise à une fonction d'activation, qui détermine la sortie de chaque neurone en fonction de ses entrées transformées. Ce traitement parallèle permet à chaque neurone de la couche cachée d'extraire des caractéristiques spécifiques à partir des mêmes données d'entrée.

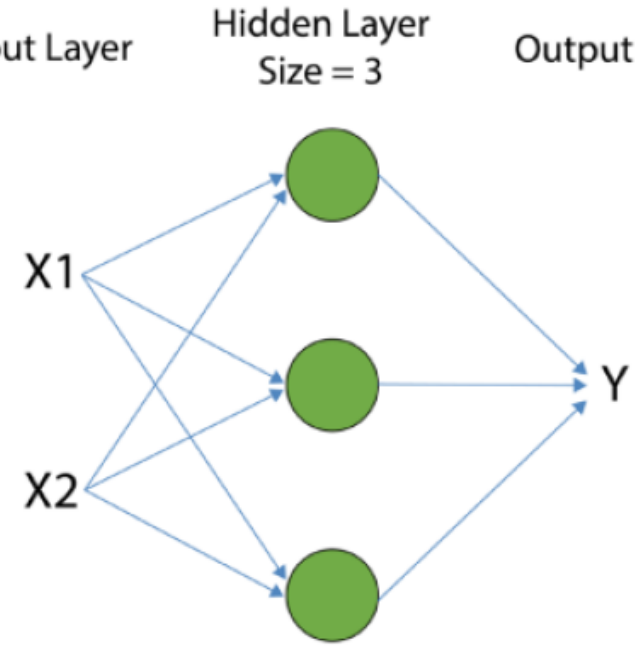


Figure 2.7 *– Représentation d'un réseau de neurones à une couche cachée avec une entrée bidimensionnelle (2D).*

Des réseaux neuronaux multicouches peuvent également être construits en empilant plusieurs couches de nuds de traitement successives. Cette architecture hiérarchique permet au réseau de modéliser des relations de plus en plus complexes et abstraites entre les données d'entrée et les sorties. Chaque couche traite les informations transmises par la couche précédente et fournit ses propres sorties en entrée de la couche suivante, créant ainsi un enchaînement de transformations progressives des données. La Figure 2.8 illustre un exemple de réseau neuronal à deux couches cachées, alimenté par un vecteur d'entrée bidimensionnel.

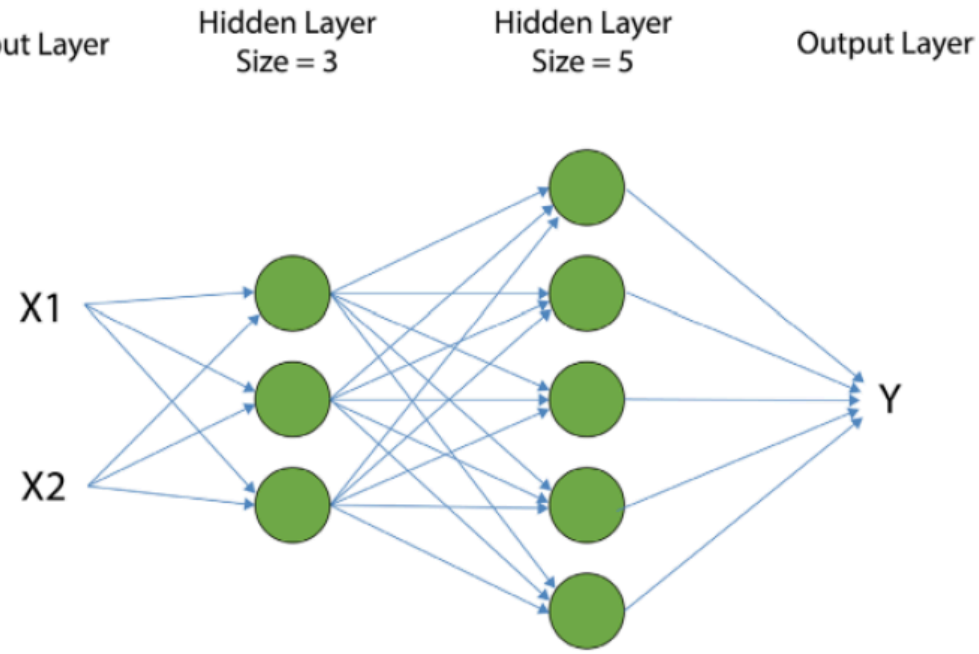


Figure 2.8 *– Représentation d'une entrée bidimensionnelle (2D) avec un réseau de neurones à deux couches cachées.*

Les Figures 2.7 et 2.8 présentent les représentations les plus courantes des réseaux neuronaux. Tout réseau neuronal est composé d'une couche d'entrée, d'une couche de sortie, ainsi que d'une ou de plusieurs couches cachées situées entre les deux. Lorsqu'un réseau ne comporte qu'une seule couche cachée, on parle alors de réseau neuronal peu profond (shallow neural network). En revanche, lorsque le réseau est constitué de plusieurs couches cachées successives, il est qualifié de réseau neuronal profond (deep neural network), cette profondeur structurelle lui conférant une capacité accrue à modéliser des relations complexes et à extraire des représentations hiérarchiques des données.

Les couches d'entrée sont généralement représentées sur la partie gauche du schéma. Dans le cas de la Figure 2.7, ces entrées correspondent aux caractéristiques X1 et X2, qui alimentent la première couche cachée constituée de trois neurones. Les flèches reliant les couches indiquent les valeurs des poids appliqués aux entrées. Dans la seconde couche cachée, les sorties issues de la première couche cachée deviennent les nouvelles entrées, poursuivant ainsi le processus de traitement hiérarchique. Les flèches reliant la première et la deuxième couche cachée symbolisent les poids synaptiques associés à ces nouvelles connexions. Enfin, la couche de sortie est généralement positionnée à l'extrémité droite de la représentation graphique ; dans le cas de la Figure 2.8, cette sortie est symbolisée par le plan noté Y, qui correspond à la valeur prédite par le réseau neuronal.

Dans l'apprentissage profond, le nombre de neurones dans la couche d'entrée est déterminé par le nombre de caractéristiques (features) des données d'entrée, tandis que le nombre de neurones dans la couche de sortie correspond aux dimensions des

données de sortie, c'est-à-dire au nombre de variables à prédire. En revanche, le choix du nombre de neurones dans les couches cachées, autrement dit la taille de ces couches intermédiaires, relève d'une décision de conception du modèle. L'utilisation de couches cachées de plus grande taille confère au modèle une flexibilité accrue, lui permettant de représenter des fonctions plus complexes et de capturer des relations non linéaires sophistiquées au sein des données. Toutefois, cette augmentation de la capacité de modélisation s'accompagne d'exigences supplémentaires en termes de quantité de données d'entraînement nécessaires, ainsi que d'une charge de calcul plus importante pour assurer l'apprentissage efficace du modèle.

Les paramètres qui doivent être spécifiés par le concepteur du modèle sont appelés hyperparamètres. Ces derniers incluent notamment le nombre de couches du réseau ainsi que le nombre de neurones par couche. Outre la configuration architecturale du réseau, d'autres hyperparamètres courants doivent également être définis, tels que le nombre d'époques (epochs) d'entraînement, c'est-à-dire le nombre de passages complets sur l'ensemble des données d'apprentissage, ainsi que le choix de la fonction de perte (loss function) qui servira de critère d'optimisation lors du processus d'apprentissage. Le réglage approprié de ces hyperparamètres joue un rôle essentiel dans la performance finale du modèle, influençant à la fois sa capacité de généralisation et sa rapidité de convergence.

Les réseaux neuronaux à propagation avant (Feedforward Neural Networks, FFNN) peuvent rencontrer des difficultés à prédire avec précision la sortie associée à une entrée donnée, en particulier lorsqu'ils sont confrontés à des problèmes complexes présentant des relations non linéaires ou des structures de données sophistiquées [73]. Dans de telles situations, il devient nécessaire d'ajuster les poids et les biais du réseau afin de réduire l'écart entre la sortie prédite et la sortie réelle. C'est précisément à ce stade qu'intervient l'algorithme de rétropropagation de l'erreur (backpropagation), qui permet de mettre à jour de manière itérative les poids et les biais du réseau en fonction des erreurs observées, améliorant ainsi progressivement la performance du modèle. Ce mécanisme d'ajustement des paramètres est illustré à la Figure 2.9.

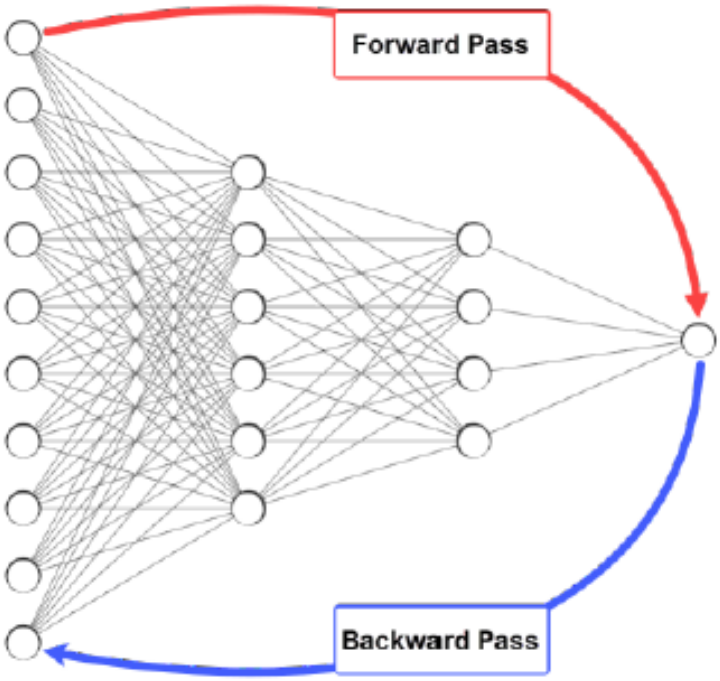


Figure 2.9 *– Forward and Backward Propagation.*

2.6.2.2 Rétro-propagation

La rétropropagation de l'erreur (backpropagation) est un algorithme d'apprentissage largement utilisé qui permet au réseau neuronal d'ajuster ses poids et ses biais au cours de l'entraînement, en propageant l'erreur depuis la couche de sortie jusqu'à la couche d'entrée (voir Figure 2.10). Ce processus repose sur le calcul des dérivées partielles de l'erreur par rapport aux poids et aux biais de chaque neurone du réseau, en appliquant la règle de dérivation en chaîne (chain rule) issue du calcul différentiel. Ainsi, l'algorithme évalue l'influence de chaque paramètre sur l'erreur globale, permettant de modifier les poids et les biais de façon à minimiser progressivement cette erreur au fil des itérations d'apprentissage [74, 75].

La rétropropagation de l'erreur est utilisée de manière répétitive au cours de la phase d'entraînement, durant laquelle le réseau est exposé à un ensemble de paires entrée-sortie. À chaque itération, les poids et les biais du réseau sont ajustés sur la base des dérivées partielles calculées, de manière à corriger progressivement les erreurs de prédiction. L'objectif principal de ce processus d'ajustement est de minimiser l'écart entre la sortie prédite par le modèle et la sortie réelle observée, améliorant ainsi la capacité du réseau à généraliser ses prédictions sur de nouvelles données non vues.

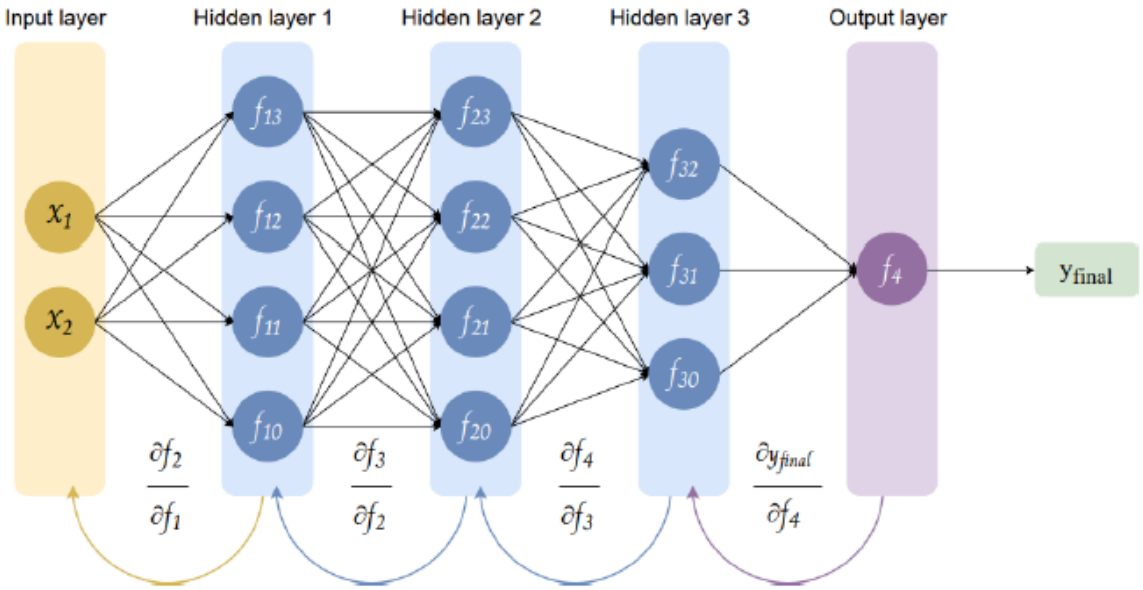


Figure 2.10 *– Backward Propagation.*

Ainsi, la rétropropagation de l'erreur permet au réseau neuronal d'apprendre et d'améliorer progressivement sa capacité à prédire avec précision la sortie correspondant à une entrée donnée. En ajustant continuellement ses paramètres en fonction des erreurs commises, le réseau affine ses représentations internes et optimise ses performances prédictives au fil des itérations d'apprentissage.

2.6.2.3 Applications du FFNNs

Les réseaux neuronaux à propagation avant (Feedforward Neural Networks, FFNN) trouvent de nombreuses applications dans divers domaines, en raison de leur capacité à apprendre des structures complexes et à effectuer des prédictions précises. Leur aptitude à modéliser des relations non linéaires et à généraliser à partir de grands volumes de données les rend particulièrement utiles pour une large variété de tâches.

- **Reconnaissance d'images et de la parole** : Les réseaux neuronaux à propagation avant (Feedforward Neural Networks, FFNN) sont couramment utilisés dans les systèmes de reconnaissance d'images [76] et de la parole [77] afin de détecter des motifs et de formuler des prédictions précises. Par exemple, dans le cadre de la reconnaissance d'images, un FFNN est capable d'apprendre à identifier des objets présents dans une image en traitant directement les valeurs des pixels comme variables d'entrée, et en extrayant progressivement des caractéristiques discriminantes au fil des couches cachées. De manière similaire, dans la reconnaissance vocale, les FFNN peuvent analyser les signaux acoustiques et en extraire des représentations adaptées permettant l'identification des mots ou des locuteurs.
- **Traitement du langage naturel** : Les réseaux neuronaux à propagation avant

(Feedforward Neural Networks, FFNN) sont également utilisés dans le domaine du traitement automatique du langage naturel (TALN) afin d'analyser et de traiter le langage humain. Par exemple, ces réseaux peuvent être employés pour effectuer des tâches de classification de textes, telles que la catégorisation de documents ou la détection de sentiments, pour extraire des informations pertinentes à partir de grandes collections de textes, ou encore pour générer des réponses automatiques dans des systèmes de dialogue et des agents conversationnels. Leur capacité à modéliser les relations complexes entre les séquences linguistiques permet d'améliorer la compréhension et la génération du langage naturel par les systèmes informatiques [78].

- **Prévisions financières** : Les réseaux neuronaux à propagation avant (Feedforward Neural Networks, FFNN) peuvent être appliqués aux prévisions financières afin de prédire l'évolution des marchés et d'aider à la prise de décision en matière d'investissement. En étant entraînés sur des données financières historiques, ces réseaux apprennent à identifier des tendances sous-jacentes, des cycles économiques ou des corrélations complexes entre différents indicateurs économiques. Une fois ces motifs appris, le modèle est en mesure de formuler des prédictions sur les évolutions futures des marchés financiers, apportant ainsi un soutien aux stratégies d'investissement, à la gestion des risques et à l'optimisation de portefeuilles [79].
- **Diagnostic médical** : Les réseaux neuronaux à propagation avant (Feedforward Neural Networks, FFNN) sont largement utilisés dans le domaine du diagnostic médical pour analyser des images médicales et assister les cliniciens dans l'établissement de diagnostics précis. Par exemple, un FFNN peut être entraîné sur un ensemble de données d'imagerie médicale, telles que des radiographies, des IRM ou des tomodensitogrammes, afin d'apprendre à détecter des signes caractéristiques de pathologies spécifiques. Grâce à leur capacité à extraire automatiquement des caractéristiques discriminantes à partir de données complexes et hétérogènes, ces réseaux permettent d'améliorer la détection précoce de maladies, d'accroître la précision diagnostique et de soutenir la prise de décision clinique [80].
- **Robotique** : Les réseaux neuronaux à propagation avant (Feedforward Neural Networks, FFNN) sont également employés dans le domaine de la robotique pour piloter des systèmes robotiques et effectuer des prédictions précises sur l'environnement. Par exemple, un FFNN peut être utilisé pour contrôler le mouvement d'un bras robotisé en lui permettant de réaliser des tâches spécifiques, telles que la préhension et la manipulation d'objets. En analysant les signaux issus de capteurs, les coordonnées spatiales ou les images captées par des caméras embarquées, le réseau neuronal peut apprendre à ajuster les mouvements du robot de manière précise et adaptative en fonction des conditions

de l'environnement, contribuant ainsi à l'automatisation de tâches complexes en milieu industriel, médical ou domestique [81].

De manière générale, les réseaux neuronaux à propagation avant (Feedforward Neural Networks, FFNN) constituent un outil à la fois polyvalent et puissant permettant de résoudre un large éventail de problèmes dans des domaines d'application variés. Leur capacité à modéliser des relations complexes, à apprendre automatiquement à partir de données et à produire des prédictions précises en fait une composante essentielle des systèmes d'intelligence artificielle modernes.

2.6.3 Réseaux neuronaux convolutifs (CNNs)

Les réseaux neuronaux convolutifs (Convolutional Neural Networks, CNN) sont des réseaux neuronaux profonds spécifiquement conçus pour les tâches de reconnaissance d'images et de vision par ordinateur [82]. Leur architecture permet d'extraire des caractéristiques hiérarchiques à partir des données d'entrée, d'une manière analogue au fonctionnement du système visuel humain. Grâce à l'application de filtres convolutifs et de couches de sous-échantillonnage, les CNN sont capables de détecter des motifs de plus en plus complexes aux différents niveaux de traitement, allant des contours élémentaires aux structures plus abstraites. Cette capacité hiérarchique d'extraction de caractéristiques leur confère une grande efficacité dans de nombreuses tâches de reconnaissance visuelle, telles que la détection d'objets, la reconnaissance faciale et la classification d'images.

2.6.3.1 Architecture des CNNs

Un réseau neuronal convolutif (Convolutional Neural Network, CNN) est composé de plusieurs types de couches successives, comprenant notamment la couche d'entrée, les couches de convolution, les couches de sous-échantillonnage (pooling layers) ainsi que les couches entièrement connectées (fully connected layers), comme illustré à la Figure 2.11. Chacune de ces couches remplit un rôle spécifique dans l'extraction progressive des caractéristiques à différents niveaux d'abstraction, permettant ainsi au réseau de traiter efficacement les structures spatiales complexes des images [83].

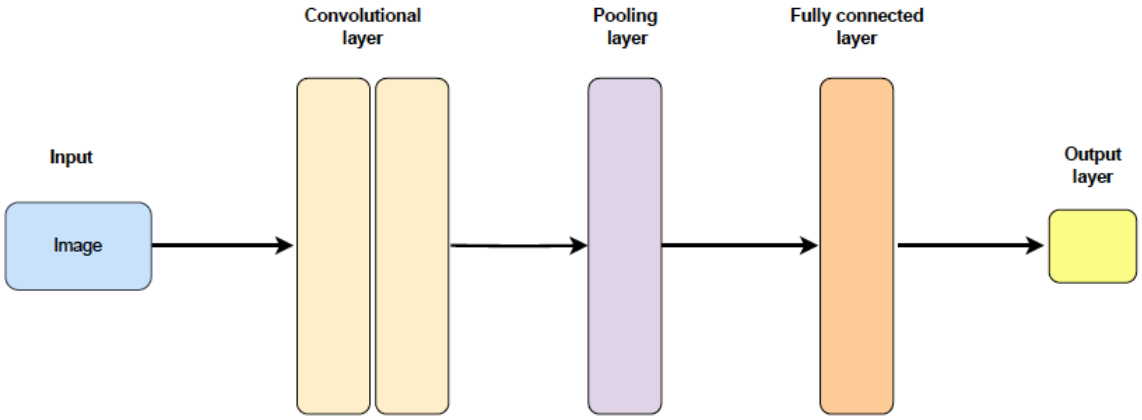


Figure 2.11 *– Couches CNN.*

- **Couche de convolution :** Cette couche applique un ensemble de filtres aux images d'entrée ou aux cartes de caractéristiques (feature maps) afin d'extraire les éléments discriminants nécessaires à la classification. Chaque filtre parcourt les données d'entrée selon un mécanisme appelé convolution, générant ainsi des cartes de caractéristiques qui mettent en évidence la présence de motifs spécifiques dans l'image, tels que les contours, les textures ou les formes géométriques. Ces filtres sont appris automatiquement au cours de l'entraînement, permettant au réseau d'extraire des représentations adaptées à la tâche de reconnaissance visuelle considérée.
- **Couche de sous-échantillonnage (Pooling Layer) :** Cette couche a pour fonction de réduire les dimensions spatiales des cartes de caractéristiques générées par la couche de convolution, tout en conservant les informations essentielles. Cette opération de réduction permet de diminuer le nombre de paramètres du modèle, de limiter le risque de surapprentissage et d'accroître l'invariance aux translations ou aux légères déformations de l'image. L'opération de sous-échantillonnage la plus couramment utilisée est le max pooling, qui consiste à sélectionner, pour chaque région définie de la carte de caractéristiques, la valeur maximale et à ignorer les autres. Ce procédé permet de conserver les activations les plus significatives, représentant les motifs les plus saillants détectés par le réseau.
- **Couche entièrement connectée (Fully Connected Layer) :** Cette couche assure la tâche de classification en exploitant les caractéristiques de haut niveau extraites par les couches de convolution et de sous-échantillonnage. Elle prend en entrée les vecteurs de caractéristiques produits par les couches précédentes et les transmet à un réseau de neurones dense, généralement implémenté sous la forme d'un perceptron multicouche (Multi-Layer Perceptron, MLP). Chaque neurone de cette couche est connecté à l'ensemble des neurones de la couche précédente, permettant ainsi une intégration globale des informations extraites.

L'entraînement de cette couche s'effectue à l'aide de l'algorithme de rétropropagation de l'erreur associé à la descente de gradient, dans le but d'ajuster les poids et biais afin de minimiser l'erreur de classification.

En complément des couches de base, les réseaux neuronaux convolutifs (Convolutional Neural Networks, CNN) peuvent également intégrer d'autres types de couches, telles que les couches de normalisation (normalization layers), les couches d'abandon (dropout layers) et les couches d'activation (activation layers), qui contribuent à améliorer leurs performances et leur robustesse. Les couches de normalisation, telles que la normalisation de lot (batch normalization), stabilisent et accélèrent le processus d'apprentissage en régulant la distribution des activations. Les couches d'abandon permettent de réduire le risque de surapprentissage en désactivant aléatoirement certaines unités durant l'entraînement, favorisant ainsi la généralisation du modèle. Enfin, les fonctions d'activation introduisent des non-linéarités essentielles au traitement des données complexes et à la capacité de représentation du réseau.

Les réseaux neuronaux convolutifs (Convolutional Neural Networks, CNN) constituent un outil puissant et flexible, largement utilisé dans les tâches de reconnaissance d'images et de vision par ordinateur à travers de nombreux domaines d'application. Leur capacité à apprendre des caractéristiques hiérarchiques directement à partir de données brutes les rend particulièrement adaptés à la résolution de problèmes complexes impliquant des structures visuelles riches. Avec l'augmentation continue de la puissance de calcul et de la disponibilité croissante de données massives, l'importance des CNNs devrait encore s'accroître à l'avenir, consolidant leur rôle central dans le développement des systèmes d'intelligence artificielle et des technologies de traitement visuel avancé.

2.6.4 Réseaux neuronaux récurrents (RNNs)

Les réseaux neuronaux récurrents (Recurrent Neural Networks, RNN) constituent une catégorie particulière de réseaux neuronaux capables d'analyser des données séquentielles, telles que les séries temporelles ou le langage naturel [84]. Contrairement aux réseaux neuronaux à propagation avant classiques, qui traitent les entrées dans une seule direction et nécessitent des tailles d'entrée et de sortie fixes, les RNNs peuvent gérer des séquences de longueur variable en maintenant un état interne qui capture l'historique des entrées précédentes. Cette capacité de modéliser des dépendances temporelles ou contextuelles les rend particulièrement adaptés à des tâches telles que la reconnaissance vocale, la traduction automatique ou encore l'analyse des sentiments, où la structure séquentielle des données joue un rôle déterminant.

2.6.4.1 Architecture des RNNs

Les réseaux neuronaux récurrents (Recurrent Neural Networks, RNN) représentent une architecture neuronale qui s'appuie sur la structure des réseaux neuronaux à propagation avant classiques (Feedforward Neural Networks, FFNN). À l'instar des FFNNs, les RNNs sont constitués d'une couche d'entrée, d'une couche de sortie, ainsi que d'une ou de plusieurs couches cachées, assurant le traitement intermédiaire de l'information [85].

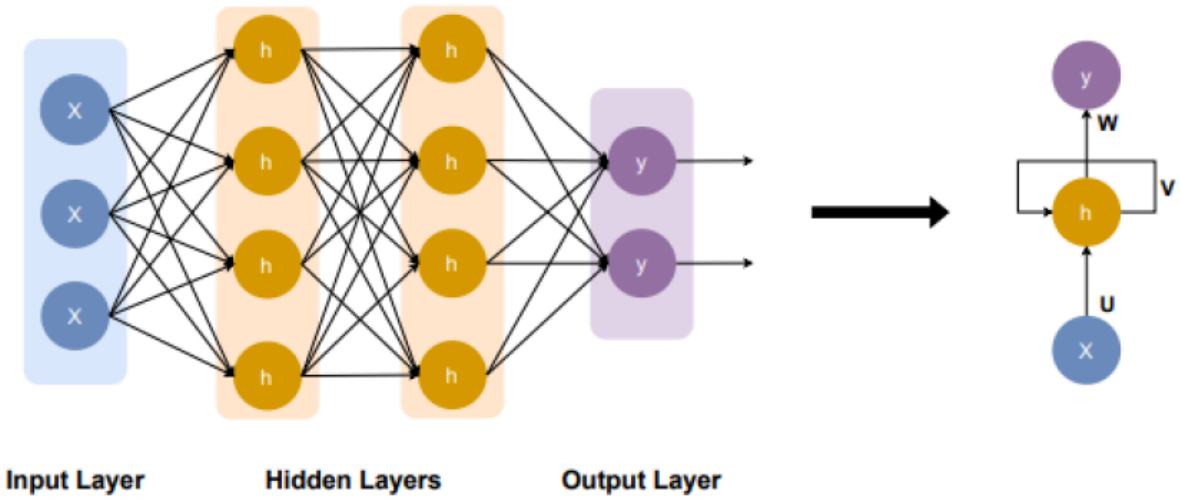


Figure 2.12 – *Propagation vers l'avant et vers l'arrière.*

Dans un réseau neuronal récurrent (Recurrent Neural Network, RNN), la couche d'entrée x traite les données d'entrée et les transmet à la couche intermédiaire h, qui peut être constituée de plusieurs couches cachées, chacune dotée de ses propres fonctions d'activation, poids et biais (voir Figure 2.12). Contrairement aux réseaux neuronaux traditionnels dépourvus de mémoire, le RNN utilise des connexions récurrentes entre les couches cachées afin de conserver une mémoire des entrées précédentes. Cette particularité permet au réseau de traiter des données séquentielles telles que les séries temporelles ou le langage naturel. À chaque pas de temps, l'entrée courante est intégrée avec l'état issu de l'entrée précédente, permettant au réseau d'accumuler l'information contextuelle au fil de la séquence. La sortie générée à chaque instant est réinjectée dans le réseau afin d'influencer les prédictions ultérieures, favorisant ainsi l'apprentissage des dépendances temporelles. Plutôt que de dupliquer plusieurs couches cachées pour chaque pas de temps, le RNN partage la même couche cachée, avec des fonctions d'activation, des poids et des biais identiques appliqués de manière itérative tout au long de la séquence.

Dans l'ensemble, les RNN se sont révélés être un outil puissant pour l'analyse des données séquentielles et ont donné des résultats impressionnants dans un large éventail d'applications.

- **Modélisation du langage et génération de texte :** Les réseaux neuronaux

récurrents (Recurrent Neural Networks, RNN) sont largement utilisés dans les tâches de traitement automatique du langage naturel (Natural Language Processing, NLP), telles que la modélisation du langage, la génération de texte ou encore l'analyse des sentiments. Leur capacité à capturer les dépendances contextuelles et séquentielles dans des données textuelles leur permet de prédire la probabilité d'apparition des mots successifs, de générer des phrases cohérentes et de comprendre les nuances émotionnelles présentes dans les textes.

- **Traduction automatique :** Les réseaux neuronaux récurrents (Recurrent Neural Networks, RNN) sont également utilisés dans les systèmes de traduction automatique, permettant de convertir des textes d'une langue vers une autre. Grâce à leur capacité à modéliser les dépendances à long terme entre les mots et à conserver le contexte global de la phrase, les RNNs permettent de produire des traductions plus fluides et grammaticalement cohérentes, en tenant compte des relations syntaxiques et sémantiques entre les différentes parties du texte.
- **Reconnaissance vocale :** Les réseaux neuronaux récurrents (Recurrent Neural Networks, RNN) sont largement exploités dans les systèmes de reconnaissance automatique de la parole (Automatic Speech Recognition, ASR) et de reconnaissance vocale. Leur aptitude à traiter des séquences temporelles leur permet d'analyser les signaux acoustiques en continu, de modéliser les dépendances temporelles entre les phonèmes et les mots, et ainsi de transcrire avec précision le langage oral en texte. Cette capacité est essentielle pour des applications telles que les assistants vocaux, les systèmes de dictée automatique et les interfaces homme-machine basées sur la parole.
- **Reconnaissance d'images :** Les réseaux neuronaux récurrents (Recurrent Neural Networks, RNN) sont également utilisés dans certaines tâches de reconnaissance visuelle, telles que la détection de visages, la reconnaissance d'objets et la reconnaissance optique de caractères (Optical Character Recognition, OCR). Bien que les réseaux convolutifs (CNNs) soient généralement privilégiés pour le traitement spatial des images, les RNNs peuvent être combinés avec ces derniers afin de modéliser des dépendances séquentielles dans des séries d'images, des séquences vidéo ou dans le cadre de la lecture de textes manuscrits, où l'ordre temporel ou spatial joue un rôle déterminant.

2.6.5 Fonctions d'activation

La fonction d'activation est utilisée afin d'introduire une non-linéarité dans la sortie de chaque neurone. Sans fonction d'activation, le réseau neuronal ne serait qu'une simple combinaison linéaire des données d'entrée, limitant considérablement sa capacité à modéliser des interactions complexes entre les variables d'entrée et de sortie. L'introduction de non-linéarités permet ainsi au réseau d'apprendre des relations

beaucoup plus riches et de traiter des problèmes à forte complexité structurelle. Il existe plusieurs types de fonctions d'activation couramment utilisées dans les réseaux neuronaux, chacune possédant des propriétés spécifiques et des avantages particuliers en fonction des tâches à réaliser. Examinons plus en détail certaines de ces fonctions d'activation et la manière dont elles sont exploitées dans les modèles d'apprentissage profond [86].

2.6.5.1 Fonction sigmoid

La fonction sigmoïde a été introduite pour la première fois dans le contexte des réseaux neuronaux par McCulloch et Pitts en 1943, puis popularisée ultérieurement par Rosenblatt dans son algorithme du perceptron en 1958. Il s'agit d'une fonction mathématique fréquemment utilisée dans les réseaux neuronaux et l'apprentissage profond. En tant que fonction d'activation, la sigmoïde transforme n'importe quelle valeur d'entrée en une valeur comprise entre 0 et 1, permettant ainsi une interprétation probabiliste de la sortie. Cette propriété de bornage est particulièrement utile dans les tâches de classification binaire et pour modéliser des probabilités dans certaines architectures de réseaux neuronaux [87].

La fonction sigmoïde est donnée par l'équation suivante :

$$\sigma(x) = \frac{1}{1 + e^{-x}} \tag{2.1}$$

où x représente la valeur d'entrée et exp désigne la fonction exponentielle. La fonction sigmoïde présente une courbe caractéristique en forme de "S" (voir Figure 2.13), et ses valeurs de sortie sont toujours comprises entre 0 et 1. Lorsque l'entrée est grande et positive, la fonction sigmoïde produit une sortie proche de 1, tandis que lorsque l'entrée est grande et négative, la sortie tend vers 0 (voir Figure 2.13). Lorsque l'entrée est égale à zéro, la fonction sigmoïde retourne une valeur de 0,5.

Aujourd'hui, la fonction sigmoïde est l'une des fonctions d'activation les plus couramment utilisées dans les réseaux neuronaux, bien que son utilisation ait diminué au profit d'autres fonctions d'activation telles que ReLU en raison de sa tendance à saturer pour les entrées importantes.

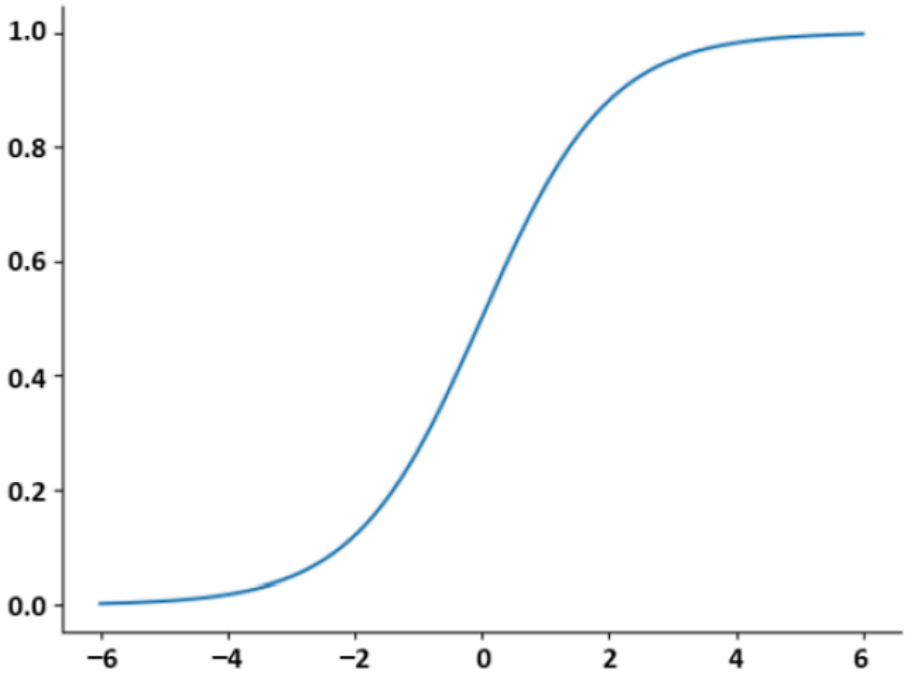


Figure 2.13 *– Fonction sigmoid.*

2.6.5.2 Fonction ReLU

La fonction ReLU (Rectified Linear Unit) est devenue l'une des fonctions d'activation les plus largement utilisées dans le domaine de l'apprentissage profond, et a démontré de bonnes performances dans de nombreuses tâches [88]. Elle a été introduite initialement par Hahnloser et al. en 2000, puis popularisée par Krizhevsky et al. lors de leurs travaux sur les réseaux de neurones profonds, notamment dans le cadre de la reconnaissance d'images à grande échelle.

La fonction ReLU (Rectified Linear Unit) est une fonction d'activation utilisée dans les réseaux neuronaux. Bien qu'elle soit d'une grande simplicité, elle s'est imposée comme une solution puissante, en raison de son efficacité pratique et de sa facilité d'implémentation dans les architectures d'apprentissage profond.

La fonction ReLU est définie comme suit :

$$f(x) = \max(0, x) \tag{2.2}$$

où x est la valeur d'entrée.

La fonction ReLU retourne la valeur d'entrée lorsque celle-ci est positive, et 0 dans le cas contraire. Cette simplicité de formulation permet un calcul rapide et efficace, tout en introduisant une non-linéarité indispensable pour modéliser des relations complexes et non linéaires dans les données (voir Figure 2.14). Grâce à ces propriétés, la ReLU favorise à la fois la convergence rapide des algorithmes d'apprentissage et la capacité de généralisation des réseaux neuronaux profonds.

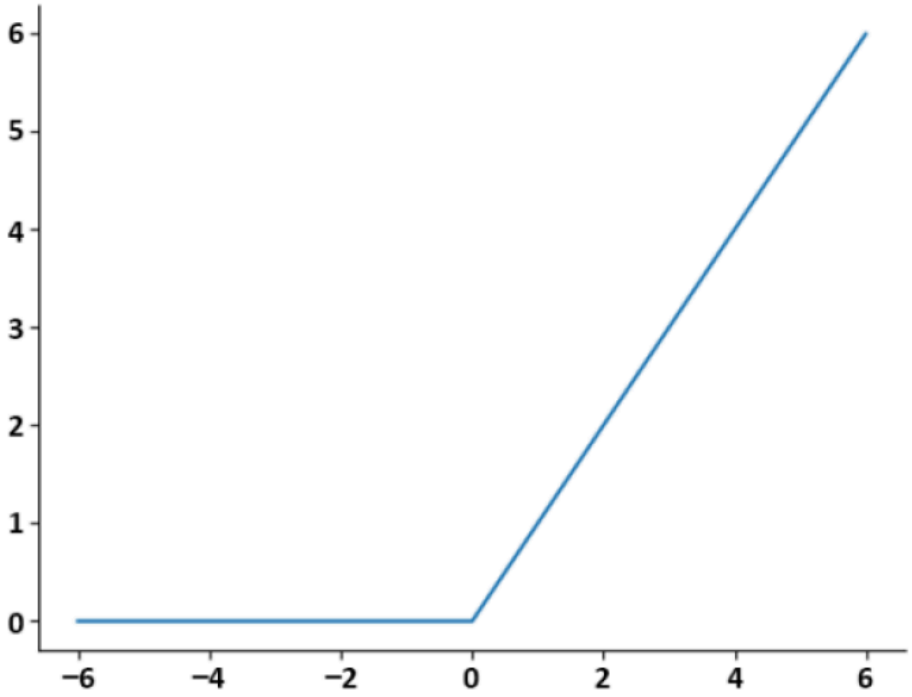


Figure 2.14 *– Fonction ReLU.*

2.6.5.3 Fonction softmax

La fonction softmax a été introduite dans le contexte des réseaux neuronaux par Bridle en 1990, puis popularisée par les travaux de Rumelhart, Hinton et Williams dans leur ouvrage de référence Parallel Distributed Processing publié en 1986. La fonction softmax est fréquemment utilisée comme fonction d'activation de sortie dans les réseaux neuronaux appliqués aux problèmes de classification multiclasse [89]. Elle permet de transformer les scores de sortie du réseau en probabilités interprétables associées à chaque classe. Par ailleurs, la fonction softmax est couramment associée à la fonction de perte par entropie croisée (cross-entropy loss), qui sert de fonction objectif dans de nombreux algorithmes d'entraînement de réseaux neuronaux.

La fonction softmax est définie comme suit :

$$softmax(x)_j = \frac{e^{x_j}}{\sum_{k=1}^{K} e^{x_k}} \tag{2.3}$$

où x est un vecteur des nombres réels, K est le nombre de classes, et j indexe la j-ième classe.

La fonction softmax prend en entrée un vecteur de K nombres réels et produit en sortie un vecteur de K probabilités, où chaque élément du vecteur de sortie correspond à la probabilité que le vecteur d'entrée appartienne à une classe donnée. Cette transformation permet de normaliser les sorties du modèle de manière à ce qu'elles soient comprises entre 0 et 1 et que leur somme soit égale à 1, rendant ainsi la fonction softmax particulièrement adaptée aux tâches de classification multiclasse.

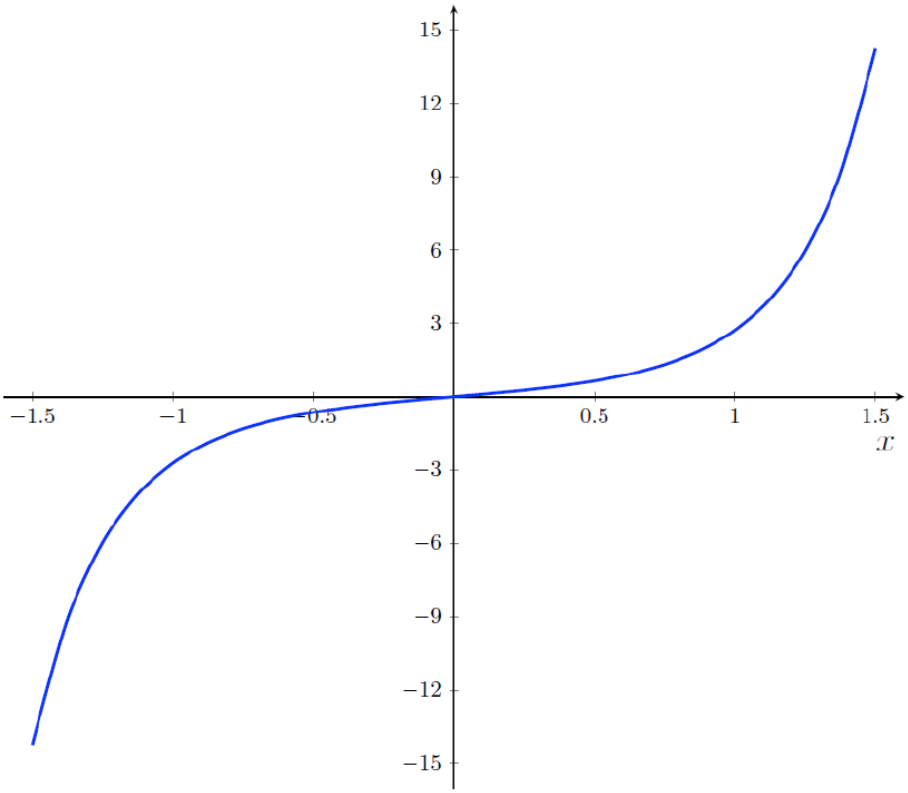


Figure 2.15 *– Fonction softmax.*

2.6.6 Fonction de perte

La fonction de perte est une fonction mathématique utilisée pour évaluer la performance d'un modèle de réseau neuronal sur une tâche donnée. Elle mesure l'écart entre la sortie prédite par le modèle et la sortie réelle, également appelée vérité terrain (ground truth). L'objectif de l'entraînement d'un réseau neuronal consiste à minimiser la valeur de cette fonction de perte, de manière à réduire progressivement l'erreur de prédiction et à améliorer la capacité du modèle à généraliser ses performances sur de nouvelles données.

Il existe différents types de fonctions de perte utilisées dans l'apprentissage profond, dont voici quelques exemples :

- Mean Squared Error (MSE) Loss
- Mean Absolute Error (MAE) Loss
- Binary Cross-Entropy Loss
- Categorical Cross-Entropy Loss
- Sparse Categorical Cross-Entropy Loss
- Hinge Loss
- Kullback-Leibler Divergence Loss
- Cosine Similarity Loss

- Triplet Loss

Le choix de la fonction de perte peut avoir un impact significatif sur les performances d'un modèle de réseau neuronal, et la sélection d'une fonction de perte appropriée constitue une étape essentielle dans la conception de l'architecture du réseau. Par exemple, dans les problèmes de régression, la fonction de perte couramment utilisée est l'erreur quadratique moyenne (Mean Squared Error, MSE), qui mesure la moyenne des carrés des écarts entre les valeurs prédites et les valeurs réelles. Dans les problèmes de classification binaire, on utilise généralement la fonction de perte par entropie croisée binaire (binary cross-entropy), tandis que pour les problèmes de classification multiclasse, la fonction d'entropie croisée catégorique (categorical cross-entropy) est fréquemment employée.

2.6.6.1 Fonctions de perte pour la classification binaire

Les fonctions de perte pour la classification binaire sont utilisées en apprentissage profond dans le cadre de problèmes où la variable de sortie est binaire, c'est-à-dire qu'elle ne peut prendre que deux valeurs distinctes, généralement 0 ou 1. Parmi les fonctions de perte couramment employées pour ce type de tâches figurent notamment la perte par entropie croisée binaire (binary cross-entropy loss) et la perte hinge (hinge loss), chacune présentant des propriétés spécifiques selon la nature du problème de classification binaire à traiter.

a. Binary Cross-Entropy Loss/LOG LOSS

Il s'agit de la fonction de perte la plus couramment utilisée dans les problèmes de classification. La perte par entropie croisée diminue à mesure que la probabilité prédite par le modèle converge vers l'étiquette réelle. Elle permet d'évaluer la performance d'un modèle de classification dont la sortie prédite correspond à une valeur de probabilité comprise entre 0 et 1. Plus la probabilité associée à la classe correcte est élevée, plus la perte est faible, ce qui traduit un meilleur ajustement du modèle aux données observées.

Lorsque le nombre de classes est de 2, il s'agit d'une classification binaire.

$$L = -\frac{1}{m}\sum_{i=1}^{m}\left(y_i \log\left(\hat{y}_i\right) + \left(1 - y_i\right)\log\left(1 - \hat{y}_i\right)\right) \tag{2.4}$$

Lorsque le nombre de classes est supérieur à 2, il s'agit d'une classification multi-classes.

$$L = -\frac{1}{m}\sum_{i=1}^{m}\left(y_i \log(\hat{y}_i)\right) \tag{2.5}$$

où y représente l'étiquette binaire réelle (0 ou 1), et $\hat{y}$ désigne la probabilité prédite pour la classe positive.

b. HINGE LOSS

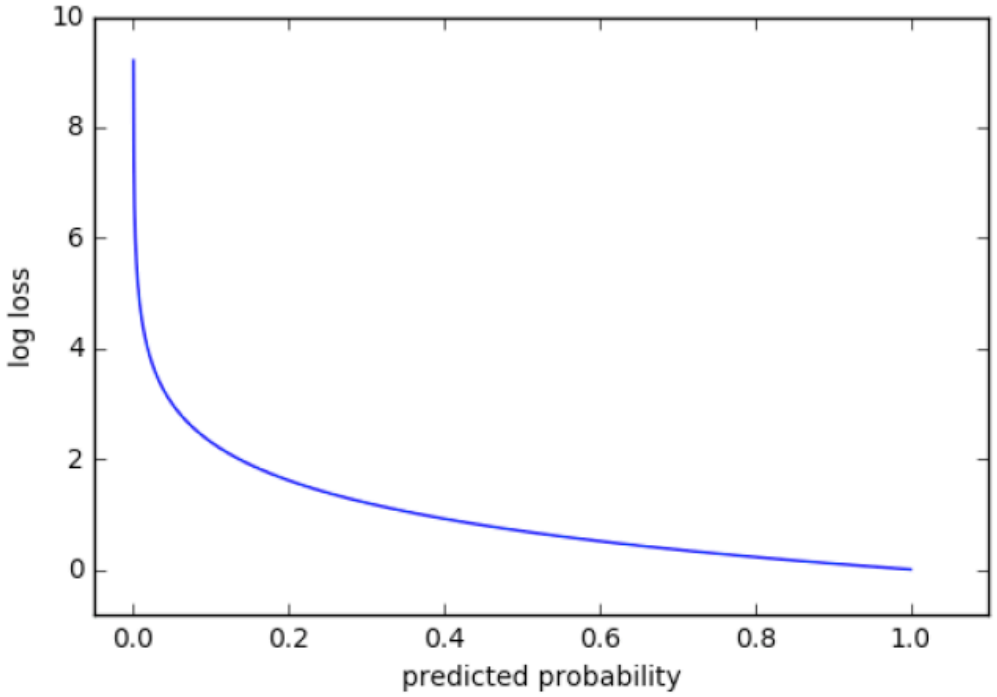


Figure 2.16 *– Log loss when true label=1.*

La perte hinge (hinge loss) peut être utilisée comme une alternative à l'entropie croisée, et a été initialement développée pour être utilisée avec l'algorithme des machines à vecteurs de support (Support Vector Machines, SVM). La perte hinge est particulièrement adaptée aux problèmes de classification dont les valeurs cibles appartiennent à l'ensemble $\{-1, 1\}$. Elle attribue une pénalité plus importante lorsque le signe de la prédiction est incorrect par rapport à la classe réelle, c'est-à-dire lorsqu'il y a discordance entre la valeur prédite et l'étiquette réelle. Cette propriété permet souvent d'obtenir de meilleures performances que l'entropie croisée dans certains scénarios de classification, en renforçant la séparation des classes lors de l'entraînement.

$$L = \max(0, 1 - yf(x)) \tag{2.6}$$

où y représente l'étiquette binaire réelle (prenant les valeurs $f(x)$ correspond au score prédit pour la classe positive, associé à l'entrée x. La fonction de perte hinge ne pénalise les erreurs de classification que lorsque le score prédit se situe du mauvais côté de la frontière de décision, c'est-à-dire lorsque $yf(x) < 1$. En revanche, lorsque la prédiction est correcte et suffisamment confiante (c'est-à-dire lorsque $yf(x) \geq 1$, la perte est nulle. Cette propriété permet au modèle de maximiser la marge entre les classes, favorisant ainsi une meilleure généralisation.

L'objectif de la fonction de perte hinge est de maximiser la marge entre la frontière de décision et les exemples d'entraînement, tout en assurant une classification correcte de ces exemples. En favorisant une séparation aussi large

que possible entre les classes, la hinge loss permet d'accroître la capacité de généralisation du modèle en réduisant sa sensibilité aux variations des données d'entraînement.

2.6.6.2 Fonctions de perte pour la classification multi-classes

La classification multiclasse correspond aux modèles prédictifs dans lesquels chaque point de données est attribué à l'une des multiples classes possibles. Dans le cadre de l'apprentissage profond, plusieurs fonctions de perte sont couramment utilisées pour traiter ce type de problèmes de classification multiclasse. Ces fonctions permettent de quantifier l'écart entre les prédictions du modèle et les classes réelles, et guident l'optimisation du modèle durant l'apprentissage.

a. Categorical Cross-Entropy Loss

La perte par entropie croisée catégorique (Categorical Cross-Entropy Loss) est couramment utilisée dans les problèmes de classification multiclasse. Elle mesure la différence entre les probabilités de classes prédites par le modèle et les étiquettes réelles codées en one-hot. La formule mathématique de la perte par entropie croisée catégorique s'exprime ainsi :

$$CCE = -\sum_{i=1}^{K} y_i log(\hat{y}_i) \tag{2.7}$$

où y représente le vecteur d'étiquettes réelles codées en one-hot, $\hat{y}$ désigne le vecteur de probabilités prédites par le modèle, et K correspond au nombre total de classes.

b. Multi-class Cross-Entropy

Le processus de one-hot encoding complique l'utilisation de l'entropie croisée multiclasse lorsque le nombre de classes et la taille des données deviennent très importants. En effet, le codage one-hot engendre des vecteurs de grande dimension, ce qui augmente la complexité computationnelle et la consommation mémoire. La sparse cross-entropy permet de contourner cette difficulté en réalisant le calcul de l'erreur directement à partir des indices de classe, sans recourir au codage one-hot. Cette approche optimise ainsi l'efficacité du calcul tout en maintenant l'exactitude des gradients nécessaires à l'apprentissage des réseaux de classification multiclasse.

$$SCCE = -\sum_{i=1}^{N} log(\hat{y}_{y_i}) \tag{2.8}$$

où y représente le vecteur des étiquettes de classes réelles et $\hat{y}$ correspond à la matrice des probabilités prédites.

c. Kullback Leibler Divergence Loss

La perte par divergence de Kullback-Leibler (KL divergence loss) calcule la divergence entre une distribution de probabilité et une distribution de référence (baseline), et mesure la quantité d'information perdue en termes de bits. Elle est utilisée comme fonction de perte afin de minimiser l'écart entre les probabilités de classes prédites par le modèle et les probabilités de classes réelles. La formule mathématique de la divergence de Kullback-Leibler s'écrit comme suit :

$$KL = -\sum_{i=1}^{K} y_i log(\frac{y_i}{\hat{y}_i}) \tag{2.9}$$

où K est le nombre de classes, y_i représente la probabilité réelle associée à la classe i, et $\hat{y}_i$ désigne la probabilité prédite pour la classe i. La perte par divergence de Kullback-Leibler mesure la différence entre la distribution réelle et la distribution prédite, et est couramment utilisée dans les modèles probabilistes afin de quantifier l'écart informationnel entre les deux distributions.

2.7 Métriques pour l'évaluation des performances des modèles d'apprentissage profond

2.7.1 Matrice de confusion

La matrice de confusion est une représentation tabulaire permettant d'évaluer les performances d'un modèle de classification pour un problème donné. Elle compare les étiquettes de classes prédites par le modèle aux étiquettes de classes réelles issues du jeu de test, en résumant les résultats sous forme matricielle. Cette représentation facilite l'analyse détaillée des performances du modèle, en mettant en évidence aussi bien les classifications correctes que les erreurs de prédiction [90].

Pour les problèmes de classification binaire, la matrice de confusion est constituée de deux lignes et de deux colonnes représentant respectivement les classes prédites et les classes réelles (voir Figure 2.17). Les colonnes correspondent aux étiquettes de classes réelles, tandis que les lignes indiquent les étiquettes de classes prédites par le modèle. Chaque cellule de la matrice exprime le nombre d'instances associées à une combinaison donnée de classe réelle et de classe prédite, permettant ainsi de quantifier avec précision les bonnes classifications ainsi que les différents types d'erreurs (faux positifs et faux négatifs).

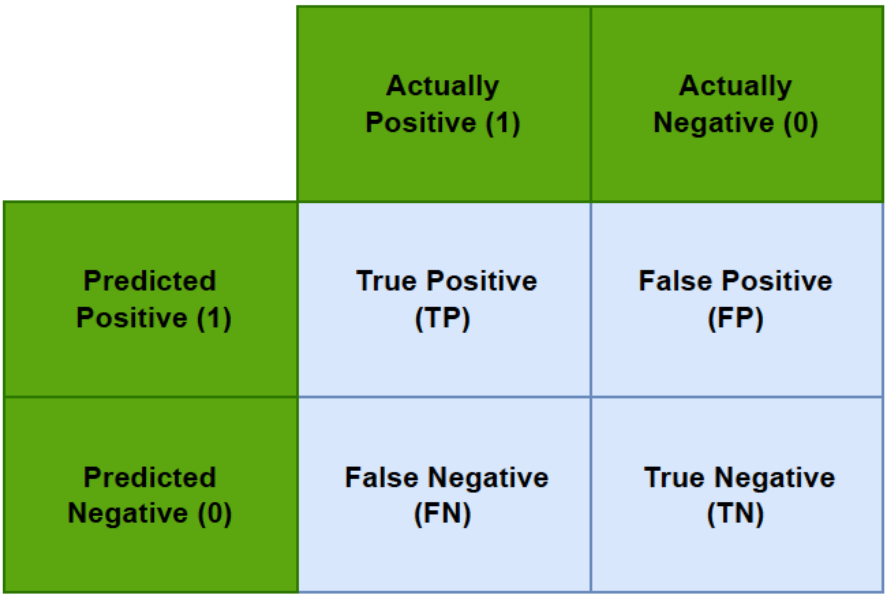


Figure 2.17 – *Matrice de confusion.*

- **Vrai Positif (VP, True Positive, TP) :** Il s'agit du nombre d'instances qui ont été correctement classées comme positives. Autrement dit, le modèle a prédit que l'instance appartenait à la classe positive, et cette prédiction correspond effectivement à la réalité.
- **Vrai Négatif (VN, True Negative, TN) :** Il s'agit du nombre d'instances qui ont été correctement classées comme négatives. Autrement dit, le modèle a prédit que l'instance appartenait à la classe négative, et cette prédiction correspond effectivement à la réalité.
- **Faux Positif (FP, False Positive) :** Il s'agit du nombre d'instances qui ont été incorrectement classées comme positives. Autrement dit, le modèle a prédit que l'instance appartenait à la classe positive, alors qu'en réalité, elle appartient à la classe négative.
- **Faux Négatif (FN, False Negative) :** Il s'agit du nombre d'instances qui ont été incorrectement classées comme négatives. Autrement dit, le modèle a prédit que l'instance appartenait à la classe négative, alors qu'en réalité, elle appartient à la classe positive.

2.7.1.1 Matrice de confusion pour la classification multi-classes

La matrice de confusion pour la classification multiclasse constitue une généralisation de la matrice de confusion binaire permettant de traiter des problèmes comportant plus de deux classes. Elle offre un cadre d'évaluation des performances d'un modèle de classification multiclasse en résumant les prédictions effectuées et les classes réelles sous la forme d'une matrice. Chaque cellule de la matrice représente le nombre d'instances associées à une classe réelle donnée qui ont été prédites comme

appartenant à une classe particulière, permettant ainsi d'identifier précisément les types d'erreurs commises par le modèle dans le cadre de multiples catégories.

Dans la classification multiclasse, la matrice de confusion comporte n lignes et n colonnes, où n représente le nombre total de classes considérées. Chaque ligne de la matrice correspond aux instances appartenant à une classe réelle donnée, tandis que chaque colonne représente les instances prédites comme appartenant à une classe spécifique. Cette organisation permet d'analyser en détail les performances du modèle pour chaque classe, en identifiant les bonnes classifications ainsi que les erreurs de confusion entre les différentes catégories.

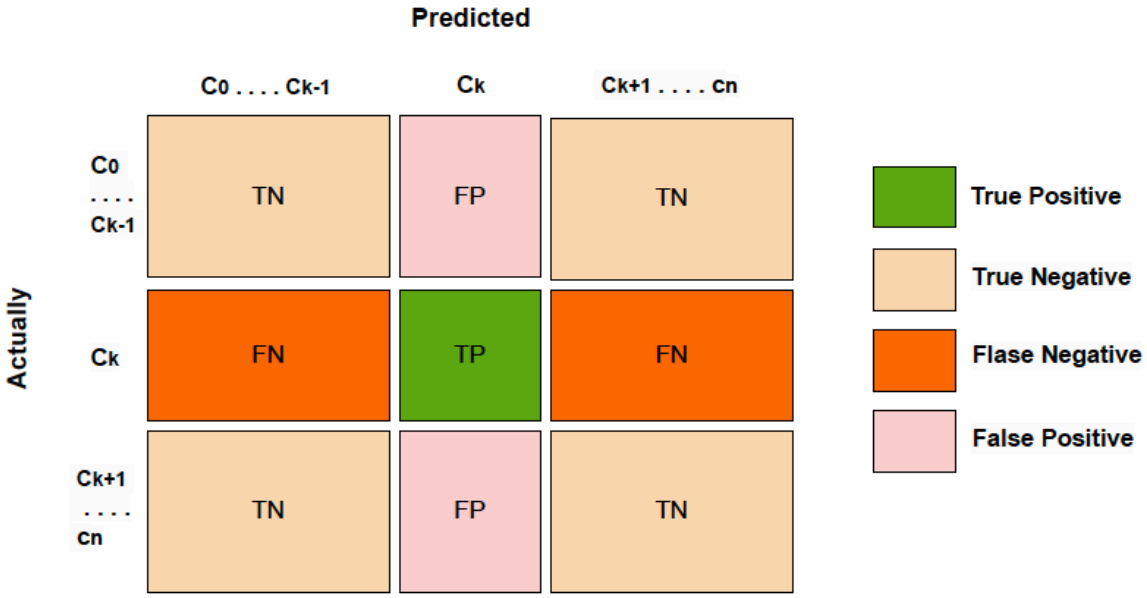


Figure 2.18 *– Matrice de confusion pour la classification multiclasse.*

2.7.2 Exactitude (Accuracy)

L'exactitude (accuracy) est l'une des métriques les plus couramment utilisées dans les problèmes de classification. Elle mesure le pourcentage de prédictions correctes réalisées par le modèle. L'exactitude se calcule en divisant la somme des vrais positifs et des vrais négatifs par le nombre total de prédictions effectuées. Cette métrique largement employée offre une évaluation globale et facilement compréhensible des performances du modèle, pouvant être aisément communiquée tant à un public technique qu'à des interlocuteurs non spécialisés. Par exemple, une exactitude de 80

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}} \tag{2.10}$$

2.7.3 Précision

La précision mesure le pourcentage de vrais positifs parmi l'ensemble des instances prédites comme positives. Elle se calcule comme le rapport entre le nombre de

vrais positifs et la somme des vrais positifs et des faux positifs. La précision constitue une métrique pertinente lorsque l'objectif est de minimiser le nombre de faux positifs, c'est-à-dire lorsque l'on souhaite s'assurer que les prédictions positives sont effectivement correctes. Elle est notamment fréquemment utilisée dans les systèmes de recherche d'information, où l'enjeu est de récupérer le plus grand nombre possible de documents pertinents tout en limitant la quantité de documents non pertinents présentés à l'utilisateur.

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \tag{2.11}$$

2.7.4 Rapel (Recall)

Le rappel, également appelé sensibilité ou taux de vrais positifs (true positive rate), est une métrique de performance permettant d'évaluer la capacité d'un modèle à identifier correctement les instances positives. Il mesure la proportion d'exemples réellement positifs que le modèle parvient à classer correctement comme positifs. Le rappel se calcule comme le rapport entre le nombre de vrais positifs et la somme des vrais positifs et des faux négatifs. Cette métrique est particulièrement pertinente dans les contextes où le coût des faux négatifs est élevé, comme dans le domaine du diagnostic médical, où une erreur de non-détection peut avoir des conséquences graves. Toutefois, le rappel présente certaines limites, dans la mesure où il ne tient pas compte des faux positifs, dont la prise en considération peut s'avérer cruciale dans d'autres types d'applications.

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{2.12}$$

2.7.5 Spécificité

La spécificité est une métrique qui évalue la capacité d'un modèle à identifier correctement les instances négatives. Elle est particulièrement utile dans les applications où le coût des faux positifs est élevé, comme dans la détection de fraudes ou le filtrage des courriers indésirables (spam). Une spécificité élevée indique que le modèle parvient efficacement à limiter le nombre de faux positifs, mais cette performance peut parfois s'accompagner d'une diminution du rappel, c'est-à-dire d'une capacité moindre à détecter l'ensemble des instances positives.

$$\text{Specificity} = \frac{\text{TN}}{\text{TN} + \text{FP}} \tag{2.13}$$

2.7.6 F1 score

F1-score est la moyenne harmonique de la précision et du rappel, calculée comme suit :

$$\text{F1 score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \tag{2.14}$$

Le score F1 constitue une métrique particulièrement appropriée lorsque la précision et le rappel revêtent une importance équivalente dans l'évaluation des performances d'un modèle. Il permet de combiner ces deux indicateurs en une seule mesure synthétique, offrant ainsi une évaluation globale de l'équilibre entre les faux positifs et les faux négatifs. Le score F1 est couramment utilisé dans les tâches de recherche d'information (information retrieval), où il est essentiel de maintenir à la fois un faible taux de faux positifs et de faux négatifs afin d'assurer la pertinence des résultats obtenus.

2.8 Conclusion

Dans ce deuxième chapitre, nous avons étudié les fondements de l'apprentissage automatique ainsi que de son sous-domaine, l'apprentissage profond. Nous avons vu comment l'apprentissage profond s'est imposé comme un outil particulièrement performant dans de nombreux domaines, grâce à la disponibilité croissante de vastes ensembles de données et à l'évolution des ressources de calcul avancées. Forts de ces connaissances théoriques, nous sommes désormais en mesure de mettre en pratique ces techniques dans des cas d'application concrets. Dans le chapitre suivant, nous présenterons la première contribution intitulée "Détection des attaques DDoS basée sur le Deep Learning à l'aide des datasets CSE-CIC-IDS2018 et Edge-IIoTset".

Chapitre 3

Détection des attaques DDoS basée sur le Deep Learning à l'aide des datasets CSE-CIC-IDS2018 et Edge-IIoTset

Sommaire

3.1 Introduction

Les progrès technologiques et la numérisation ont apporté d'immenses avantages à la société, avec des innovations telles que le calcul en périphérie (edge computing), la blockchain, la robotique et l'Internet des Objets (IoT), qui transforment en profondeur de nombreux aspects de la vie [91]. Toutefois, bien que ces technologies émergentes offrent de multiples bénéfices, leur connectivité accrue et leur rôle central dans les systèmes critiques élargissent également la surface d'attaque exploitable par des acteurs malveillants. À mesure que la dépendance aux services et opérations connectés s'accroît, tant au sein des entreprises que des infrastructures, les vulnérabilités exploitables par des attaquants se multiplient. Ces derniers explorent en permanence ces failles pour mener des activités malveillantes, exposant ainsi individus et organisations à des risques majeurs [92].

Parmi la diversité des cyberattaques, les attaques par déni de service distribué (DDoS) se sont imposées comme une menace dominante et persistante depuis leur apparition initiale en 1974 [93]. Les attaquants DDoS mobilisent un réseau d'appareils compromis, connu sous le nom de botnet, où chaque appareil infecté, ou bot, participe au déluge de trafic dirigé vers la cible, rendant ainsi cette dernière inaccessible aux utilisateurs légitimes. Les attaques DDoS peuvent avoir des répercussions considérables sur les entreprises et organisations, perturbant les opérations, entraînant des pertes financières et portant atteinte à leur réputation. Dans certains cas, ces attaques ont même été utilisées pour provoquer des dommages physiques [91].

En 2018, Google a subi une attaque DDoS massive qui a perturbé ses services durant plusieurs heures. Les pirates ont utilisé une technique dite de ń Memcached Amplification ż, exploitant des serveurs Memcached mal configurés pour générer de très grandes réponses à des requêtes simples, submergeant ainsi le serveur cible. Cette attaque a été considérée, à l'époque, comme la plus importante attaque DDoS jamais enregistrée [94]. En février 2021, Amazon a également été victime d'une attaque DDoS majeure, perturbant ses services pendant plusieurs heures. Les attaquants ont utilisé une technique dite de ń Connection Flood ż, qui consiste à envoyer simultanément un très grand nombre de requêtes de connexion à un serveur, le saturant et le rendant inaccessible. Amazon a réagi rapidement en bloquant l'adresse IP de l'attaquant, mais l'incident a néanmoins eu un impact négatif sur ses services [95].

Les techniques traditionnelles de détection fondées sur des règles ont été largement utilisées pour détecter les attaques DDoS. Cependant, ces méthodes ont montré leurs limites en raison de leur manque d'adaptabilité face aux nouvelles techniques d'attaque et de leur forte sensibilité aux faux positifs. De plus, les attaquants ont développé des techniques sophistiquées permettant de dissimuler leurs attaques, rendant leur détection encore plus complexe. Face aux insuffisances de ces méthodes, la recherche de techniques de détection plus intelligentes a conduit au développement de l'apprentissage profond (deep learning). Les algorithmes d'apprentissage profond

jouent un rôle crucial dans la lutte contre l'évolution permanente des attaques DDoS. Les défenses traditionnelles rencontrent des difficultés à détecter et à réduire efficacement ces attaques en raison de la rapidité d'évolution des techniques d'attaque et de l'ampleur des attaques facilitées par les botnets. En exploitant la puissance des réseaux de neurones artificiels, les algorithmes d'apprentissage profond analysent d'importants volumes de données afin d'identifier des schémas d'attaque subtils et complexes. En outre, ils sont capables de s'adapter continuellement aux nouvelles techniques d'attaque, améliorant ainsi l'efficacité de la défense au fil du temps. Cette approche adaptative et robuste offre une défense plus efficace et proactive contre les attaques DDoS sophistiquées.

L'objectif de cette étude est de proposer différentes approches de systèmes de détection basés sur l'apprentissage profond, spécifiquement conçus pour les attaques DDoS. Pour ce faire, nous utilisons le jeu de données bien connu CSE-CIC-IDS2018 afin de créer un sous-ensemble dédié aux attaques DDoS. Le but principal est de développer un système de détection robuste, basé sur le deep learning, capable d'identifier et d'atténuer avec précision divers types d'attaques DDoS.

Dans ce chapitre, nous proposons deux modèles pour la classification des attaques DDoS : un modèle de réseau neuronal profond (Deep Neural Network, DNN) et un modèle de réseau de neurones convolutifs (Convolutional Neural Network, CNN). Le modèle DNN utilise des techniques de classification binaire, multiclasses (avec encodage des étiquettes et encodage one-hot) ainsi qu'une classification multi-étiquettes. Les expérimentations avec le DNN ont été réalisées sur le jeu de données CSE-CIC-IDS2018. Le modèle CNN a été testé uniquement pour la classification multiclasses sur le jeu de données Edge-IIoTset. Nous évaluerons la validité des modèles DNN et CNN en comparant leurs performances aux travaux antérieurs.

Lors de la phase d'apprentissage, les modèles DNN et CNN apprendront à partir du jeu de données annoté, en adaptant leurs paramètres internes à travers un processus d'optimisation itératif. Nous ajusterons avec soin les hyperparamètres des deux modèles afin d'améliorer leurs performances dans la détection et la classification précise des différents types d'attaques DDoS. Pour évaluer l'efficacité des modèles proposés, nous utiliserons des métriques d'évaluation rigoureuses telles que la précision (accuracy), la précision au sens strict (precision), le rappel (recall) et le score F1 (F1-score).

3.2 Revue de la littérature

Dans les études récentes sur la détection d'intrusions, les méthodes basées sur l'apprentissage automatique et l'apprentissage profond ont été largement exploitées. Des systèmes de détection d'intrusions de plus en plus efficaces sont en cours de développement dans ce domaine grâce à l'évolution des algorithmes d'apprentissage

automatique et d'apprentissage profond, soutenue par l'accès aux mégadonnées. Dans cette section de la littérature, nous présentons quelques-unes des études récentes portant sur les attaques DDoS.

Di Mauro et al. [95] ont proposé une revue des différentes techniques de sélection de caractéristiques et de leurs performances sur les ensembles de données couramment utilisés, résumant de nombreuses méthodes et comparant leurs résultats sur un ensemble de données d'attaques DDoS. Ils ont analysé le temps de sélection des caractéristiques, le temps d'entraînement et les matrices de corrélation. Leur travail a permis de démontrer les performances de différentes techniques selon les types d'attaques et les jeux de données, en fonction des caractéristiques extraites des données DDoS. De manière générale, la conception des systèmes de détection d'intrusions existants repose sur diverses approches d'apprentissage automatique permettant d'identifier différentes catégories d'attaques. Les comparaisons mettent en évidence les forces et faiblesses relatives de ces méthodes. Les techniques hybrides combinant plusieurs algorithmes sont souvent préférées aux méthodes uniques. Le prétraitement par sélection de caractéristiques est fréquemment intégré afin de réduire la charge de traitement et d'améliorer les performances. Étant donné la diversité des algorithmes d'extraction de caractéristiques existants, des expérimentations sont nécessaires pour identifier les techniques optimales adaptées à la structure des données. Globalement, des évaluations rigoureuses permettent de déterminer l'approche de sélection de caractéristiques la plus appropriée dans le cadre de la conception d'un IDS donné.

Kamalov et al. [96] ont appliqué une méthode de fusion en apprentissage automatique pour développer un nouveau modèle IDS. Pour identifier les caractéristiques pertinentes dans le jeu de données, les auteurs ont utilisé la technique de décomposition orthogonale de la variance. Les attributs sélectionnés ont servi à construire un réseau neuronal profond pour la détection d'intrusions. La technique proposée atteint une précision de détection de 100 % pour les attaques DDoS. Wei et al. [97] ont proposé un modèle hybride intégrant deux algorithmes basés sur l'apprentissage profond pour la détection et la classification réussies des attaques DDoS. En identifiant automatiquement les ensembles de caractéristiques les plus significatives, le composant Autoencoder du modèle réalise une extraction efficace des caractéristiques. Pour réduire la surcharge de traitement face aux multiples catégories d'attaques DDoS, le réseau de neurones perceptron multicouche (MLP) exploite des ensembles de caractéristiques compressés et réduits. Les résultats de test ont démontré que la méthode proposée atteint des scores de précision et de F1 supérieurs à 98

Odumuyiwa et al. [98] ont utilisé des techniques d'apprentissage automatique non supervisé pour classer les paquets réseau au niveau de la couche transport afin d'identifier les attaques DDoS dans les réseaux IoT. Cette étude a développé indépendamment deux algorithmes d'apprentissage profond et deux algorithmes de clustering pour atténuer les attaques DDoS. Les attaques par exploitation telles que les inondations SYN du protocole TCP et les attaques UDP lag ont été mises en évi-

dence. Durant la phase expérimentale, les algorithmes ont été entraînés sur les jeux de données Mirai, BASHLITE et CIC-DDoS2019. Les résultats ont montré que l'autoencodeur a obtenu les meilleures performances globales avec la plus haute précision sur l'ensemble des jeux de données.

Cil et al. [99] ont proposé un modèle basé sur les réseaux de neurones profonds pour détecter les attaques DDoS à partir d'échantillons de paquets issus du trafic réseau. Le modèle DNN fonctionne efficacement sur le jeu de données CIC-DDoS2019 grâce à l'intégration d'algorithmes d'extraction de caractéristiques et de classification dans sa structure, avec des couches qui s'ajustent automatiquement durant l'apprentissage.

Amaizu et al. [100] ont développé un système complet et performant de détection des attaques DDoS pour les réseaux 5G et B5G (5G Beyond). Le système proposé combine un perceptron multicouche composite avec une méthode efficace d'extraction de caractéristiques pour identifier et classifier les types d'attaques DDoS. Les simulations et les tests sur les jeux de données ont montré une précision de détection de 99,66 % et une perte de 0,011. Les performances du cadre de détection proposé ont également été comparées à celles d'autres chercheurs.

Afin de pallier certaines limites et introduire une nouvelle taxonomie des attaques DDoS, incluant une classification basée sur les flux réseau, Khempetch et al. [101] ont proposé le jeu de données CIC-DDoS2019. Une architecture DNN et LSTM est utilisée pour l'identification des attaques DDoS.

Hussain et al. [102] ont proposé une méthode reposant sur l'architecture de deep learning ResNet18 pour identifier les attaques DoS et DDoS dans les systèmes IoT. Les caractéristiques du trafic réseau ont été transformées en représentations d'images servant d'entrée à l'entraînement de ResNet18. Le modèle a été entraîné pour classifier 11 types d'attaques et le trafic bénin.

Jia et al. [103] ont présenté des modèles basés sur la convolution et LSTM pour détecter les attaques DDoS en fonction des fluctuations de trafic dans les réseaux IoT. Ils ont combiné le jeu de données CICDDoS2019 avec un ensemble de données généré à l'aide des simulateurs DDoS BoNeSi et SlowHTTPPTest. Ils ont évalué l'adéquation des modèles pour les réseaux IoT ainsi que leurs performances.

Shurman et al. [104] ont proposé des méthodes hybrides et basées sur le deep learning pour détecter les attaques DoS et DrDoS dans les réseaux IoT en intégrant des approches par signatures et par anomalies. Leurs modèles, tous fondés sur des réseaux LSTM, ont été entraînés et testés sur le jeu de données CIC-DDoS2019.

Li et al. [105] ont suggéré une stratégie en trois étapes pour contrer les attaques DDoS visant les dispositifs IoT : accélération du calcul de l'entropie, détection précoce et optimisation des résultats de détection. Les expériences ont utilisé les jeux de données DARPA 1999, DARPA DDoS et UNB CIC-DDoS2019. Leur solution se distingue par une faible latence et de hautes performances, facilitant ainsi son intégration en temps réel dans les systèmes de défense IoT.

Sharafaldin et al. [106] ont examiné les jeux de données DDoS existants et leurs

lacunes, proposant ainsi un nouveau jeu de données pour évaluer les IDS et IPS à travers un banc d'essai intégrant réseaux d'attaque et de victime. Leur jeu de données CIC-DDoS2019 constitue une amélioration par rapport aux ensembles existants.

Javeed et al. [107] ont proposé une architecture compatible SDN utilisant des algorithmes hybrides de deep learning pour détecter les cybermenaces tout en tenant compte des contraintes de ressources des dispositifs IoT. L'approche a été validée sur le jeu de données CIC-DDoS2019.

Alamri et Thayananthan [108] ont développé un schéma de détection des attaques DDoS ciblant les réseaux SDN. Les violations de seuil sont détectées par surveillance de la bande passante, déclenchant l'algorithme Extreme Gradient Boosting (XGBoost) afin de différencier trafic normal et malveillant. L'approche a été testée sur les jeux de données CIC-DDoS2019, NSL-KDD et CAIDA.

Pour prévenir les attaques DDoS, De Assis et al. [109] ont proposé un système de défense pour les réseaux SDN, fréquemment utilisés dans l'IoT. Ils ont évalué différentes architectures d'apprentissage telles que MLP, CNN, D-MLP et LR, en classifiant les flux comme DDoS ou normaux. Des flux IP simulés provenant du jeu de données CIC-DDoS2019 ainsi que du trafic SDN généré ont été utilisés pour l'entraînement et les tests.

Zhijun et al. [110] ont étudié les mécanismes des attaques DDoS et développé une nouvelle approche de détection exploitant de multiples caractéristiques et l'algorithme Factorization Machine (FM). En extrayant des caractéristiques informatives des règles de flux réseau, ils ont ciblé la détection des attaques DDoS à faible débit. Les expérimentations ont démontré une précision de détection de 95,80 %, mettant en évidence l'efficacité de leur modèle FM dans l'identification de ces menaces insidieuses.

Yang et al. [111] ont développé un cadre basé sur les réseaux SDN pour détecter et atténuer les attaques DDoS à l'aide de techniques d'apprentissage automatique. Le cadre comporte trois modules : collecte des flux réseau, identification des attaques DDoS par apprentissage automatique, et distribution de règles de flux mises à jour pour contrer les attaques détectées. L'approche a été validée sur le jeu de données KDD99.

Sudar et al. [112] ont proposé un cadre basé sur les flux et l'apprentissage automatique pour détecter et atténuer les attaques DDoS à faible débit (LR-DDoS). Des caractéristiques clés extraites des flux de trafic réseau ont servi d'entrée à trois modèles : SVM, arbre de décision C4.5 et Naive Bayes. Le SVM a obtenu les meilleures performances.

Alashhab et al. [113] ont proposé une méthode de détection des attaques LDDoS basée sur un modèle de deep learning utilisant des fonctions d'activation LSTM. Leur modèle analyse les caractéristiques des attaques LDDoS et du trafic normal dans les réseaux IoT. Les expérimentations ont montré une précision de détection de 98,88

3.3 Détection d'intrusion basée sur les réseaux neuronaux profonds à l'aide du dataset CSE-CIC-IDS2018

L'objectif de cette section est de présenter la mise en uvre de modèles de réseaux de neurones profonds (DNN) pour la détection des attaques DDoS en utilisant différentes approches et techniques d'apprentissage profond. Nous avons également évalué les performances de ces modèles en termes de précision (accuracy), de précision au sens strict (precision), de rappel (recall) et de score F1 (F1-score). En exploitant les techniques d'apprentissage profond, ce travail vise à renforcer la robustesse de l'infrastructure réseau face aux activités malveillantes. Afin d'atteindre cet objectif, nous avons utilisé le jeu de données CSE-CIC-IDS2018, qui constitue un ensemble de données complet et largement utilisé dans le domaine de la sécurité des réseaux. Cette section présente l'architecture réseau utilisée dans notre expérimentation de détection et de classification des attaques DDoS.

3.3.1 Dataset

Le jeu de données sélectionné est le CSE-CIC-IDS2018 [114], [115]. Ce jeu de données a été créé dans le cadre d'un projet collaboratif entre le Communications Security Establishment (CSE) et le Canadian Cybersecurity Institute (CIC), qui utilise des profils pour générer de manière systématique des jeux de données en cybersécurité. Il fournit une description détaillée des intrusions, ainsi que des modèles de distribution abstraits pour les applications de bas niveau, les protocoles et les entités du réseau. Ce jeu de données se compose de dix fichiers CSV représentant dix jours de capture de flux réseau, contenant plus de 16,2 millions d'échantillons. De plus, l'outil CICFlowMeter a permis d'extraire plus de 80 caractéristiques. Ce jeu de données englobe six types principaux d'attaques d'intrusion : le déni de service distribué (DDoS), le déni de service (DoS), le Botnet, la force brute (Brute Force), l'infiltration et les attaques web.

3.3.2 Prétraitement des données

Lors du traitement d'un jeu de données, il est essentiel de garantir sa qualité, son organisation ainsi que la clarté des données avant de l'utiliser pour l'entraînement d'un modèle d'apprentissage profond. Les réseaux de neurones profonds nécessitent généralement de vastes volumes de données pour leur apprentissage, ce qui peut engendrer divers problèmes : données inutiles, redondantes, erronées, manquantes (valeurs NaN Not A Number), infinies, ainsi que d'autres anomalies.

Le prétraitement des données constitue une étape cruciale afin de s'assurer que les données fournies en entrée du modèle d'apprentissage profond soient propres, co-

hérentes et pertinentes par rapport à la tâche de classification ou de prédiction visée. Cette phase permet d'améliorer la qualité des données, de les rendre plus homogènes et de les adapter aux exigences du modèle à concevoir. Le prétraitement des données peut également contribuer à réduire les temps d'entraînement et de test du modèle. En s'assurant que les données soient nettoyées et prêtes à être utilisées pour l'entraînement, on garantit que les performances du modèle ne soient pas affectées par des données manquantes, erronées ou redondantes. Le prétraitement des données représente ainsi une étape fondamentale pour assurer la fiabilité et l'efficacité du modèle d'apprentissage profond.

Les étapes de prétraitement des données pour l'apprentissage profond peuvent inclure différentes techniques telles que la normalisation des données, la suppression des valeurs infinies (NaN ou Inf), la gestion des données manquantes, l'élimination des doublons, la suppression des données inutiles, ainsi que le codage des variables et autres transformations nécessaires.

3.3.2.1 Nettoyage des données

Le nettoyage des données constitue une étape essentielle permettant de disposer d'un jeu de données propre et fiable, condition indispensable pour réaliser des analyses et des modélisations plus précises. Un ensemble de données contenant des erreurs, des redondances ou des informations manquantes peut compromettre la performance et la robustesse des modèles d'apprentissage profond, en particulier dans le domaine sensible de la détection des intrusions et des attaques DDoS.

Après une analyse approfondie du jeu de données, nous avons constaté un taux de valeurs manquantes d'environ 1,1 %. Afin de garantir l'intégrité et la qualité des données exploitées pour l'entraînement des modèles, nous avons procédé à une opération de nettoyage systématique. Dans un premier temps, certaines caractéristiques ont été éliminées, notamment les attributs "Timestamp", "Flow ID", "Src IP", "Src Port" et "Dst IP", car ces variables sont directement liées aux informations de contact réseau et n'apportent pas d'informations significatives sur les caractéristiques intrinsèques des attaques. Leur présence pourrait même introduire du bruit ou des biais dans l'apprentissage du modèle, en particulier si l'objectif est de détecter des patterns d'attaques génériques et indépendants des identifiants de flux spécifiques.

Par la suite, nous avons supprimé toutes les valeurs représentant des données infinies (Inf), des valeurs manquantes (NaN) ainsi que les lignes dupliquées. L'élimination de ces éléments redondants et aberrants contribue non seulement à améliorer la qualité globale des données, mais aussi à accélérer le processus d'apprentissage et à éviter les biais ou erreurs lors de l'entraînement du modèle de classification.

L'ensemble de ces opérations de nettoyage des données vise donc à garantir que le jeu de données soit conforme aux exigences des algorithmes d'apprentissage profond, en minimisant l'impact de données erronées ou non pertinentes sur les performances

globales du système de détection.

3.3.2.2 Codage des données

Dans certains cas, les variables catégorielles peuvent s'avérer difficiles à exploiter directement dans les analyses statistiques ou au sein des algorithmes d'apprentissage automatique. En effet, de nombreux algorithmes de machine learning, et plus particulièrement ceux utilisés dans l'apprentissage profond, nécessitent que les variables d'entrée soient de nature numérique afin de pouvoir effectuer les calculs d'optimisation et d'apprentissage sur des représentations mathématiques des données.

Pour cette raison, il est indispensable de transformer les variables catégorielles en une forme numérique appropriée. Afin de réaliser cette transformation, nous avons utilisé la méthode str.replace() de la bibliothèque Pandas. Cette méthode nous a permis de remplacer les différentes modalités catégorielles par des valeurs entières, facilitant ainsi leur interprétation numérique par les algorithmes. Chaque valeur catégorielle a ainsi été associée à un identifiant entier unique, permettant de conserver l'information tout en la rendant compatible avec les modèles d'apprentissage.

Après cette première opération de transformation, nous avons utilisé la méthode astype() afin de convertir le type de données en entiers codés sur 64 bits (int64). Cette étape était indispensable pour assurer la compatibilité des colonnes concernées avec certains algorithmes de machine learning qui exigent un format numérique standardisé et optimisé pour le calcul.

Grâce à l'application combinée de ces deux méthodes, nous avons pu convertir efficacement la colonne "Label" en une variable numérique adaptée à nos analyses et aux modèles d'apprentissage automatique utilisés par la suite [116].

3.3.2.3 Normalisation et standardisation

La normalisation permet de ramener l'ensemble des variables quantitatives à une même échelle de valeurs, généralement comprise entre 0 et 1. Cette transformation facilite grandement l'apprentissage des algorithmes de machine learning en supprimant les différences d'échelle entre les variables, ce qui permet aux modèles d'accorder une importance équivalente à chaque caractéristique lors de la phase d'optimisation des paramètres [117].

De son côté, la standardisation constitue une méthode d'échelle différente qui transforme les données afin qu'elles suivent une distribution gaussienne (normale) de moyenne nulle et d'écart-type égal à un. Ce procédé centre les variables autour de zéro et ajuste leur dispersion, ce qui est particulièrement adapté aux algorithmes sensibles aux écarts de variance, tels que les réseaux de neurones profonds (DNN), les régressions ou les modèles de classification linéaires [118].

Dans le but d'améliorer les performances, la stabilité et la fiabilité de notre modèle de réseau de neurones profonds, ainsi que de favoriser une convergence plus

rapide durant l'entraînement, nous avons appliqué la méthode de standardisation sur les données d'apprentissage et de test. Pour ce faire, nous avons utilisé la fonction StandardScaler du module sklearn.preprocessing. Cette méthode constitue une étape fondamentale du prétraitement des données, car elle rend les différentes variables comparables, facilite la convergence des algorithmes d'optimisation et contribue à une meilleure généralisation du modèle en réduisant les biais liés aux écarts d'échelle entre les variables d'entrée.

Ainsi, la standardisation s'est avérée être une opération incontournable dans le cadre de notre préparation des données avant l'entraînement du modèle DNN.

3.3.2.4 Fractionnement des données

En apprentissage profond, les performances d'un modèle ne doivent jamais être évaluées sur les mêmes données que celles utilisées pour l'entraînement. Lors de l'entraînement d'un modèle de deep learning, il est essentiel de constituer deux sous-ensembles distincts de données : un jeu de données d'apprentissage et un jeu de données de test. Le jeu de données d'apprentissage sert à ajuster les paramètres internes du modèle, tandis que le jeu de données de test permet d'évaluer les performances du modèle sur des données inédites, afin de mesurer sa capacité de généralisation.

De manière générale, la répartition classique consiste à allouer 80 % des données à la phase d'apprentissage et 20 % aux tests de validation des performances (voir Figure 3.1).

Pour réaliser cette séparation des données en Python, nous utilisons la fonction train_test_split() appartenant au module model_selection.

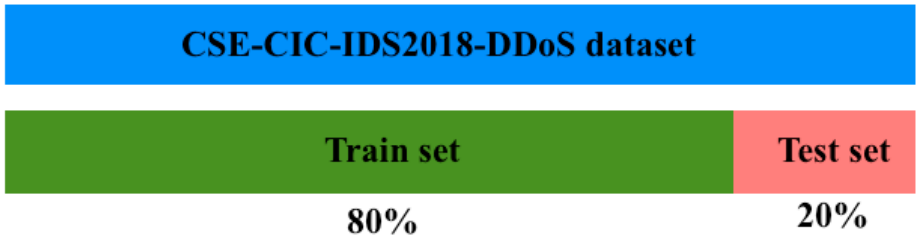


Figure 3.1 *– Division du dataset CIC-IDS2018 en deux parties.*

3.3.3 Création du modèle

Différentes méthodes de classification sont utilisées pour détecter les attaques DDoS, telles que la classification binaire, la classification multiclasses (avec à la fois l'encodage des étiquettes label encoding et l'encodage one-hot) ainsi que la classification multi-étiquettes. À cet effet, nous avons construit un modèle de réseau de neurones multicouche (Multilayer Perceptron [119]) comportant quatre couches cachées de type Dense : la première couche comprend 128 neurones, la deuxième 64

neurones, et les troisième et quatrième couches comportent chacune 32 neurones. Les couches cachées utilisent la fonction d'activation ReLU (Rectified Linear Unit) [120], tandis que la couche de sortie utilise la fonction d'activation sigmoid pour la classification binaire et la classification multi-étiquettes, et la fonction d'activation softmax pour la classification multiclasses. Le choix de la fonction d'activation de la couche de sortie dépend donc de l'approche de classification adoptée.

Afin de limiter le risque de surapprentissage (overfitting), le modèle a été régularisé en appliquant une régularisation de type L2 [121], avec un coefficient lambda fixé à 0,0001 sur chaque neurone. Le modèle a été compilé en utilisant l'optimiseur Adamax. Le choix de la fonction de perte (loss function) dépend du type de classification employé : categorical_crossentropy pour la classification multiclasses et binary_crossentropy pour la classification binaire et multi-étiquettes. L'évaluation du modèle est réalisée à l'aide de la métrique accuracy.

L'entraînement du modèle est effectué avec une taille de lot (batch size) de 128 et sur un nombre d'époques fixé à 30. Les données sont divisées en 90 % pour l'entraînement et 10 % pour la validation. Enfin, le modèle est évalué sur l'ensemble de test. La Figure 3.2 illustre le système proposé de détection des attaques DDoS.

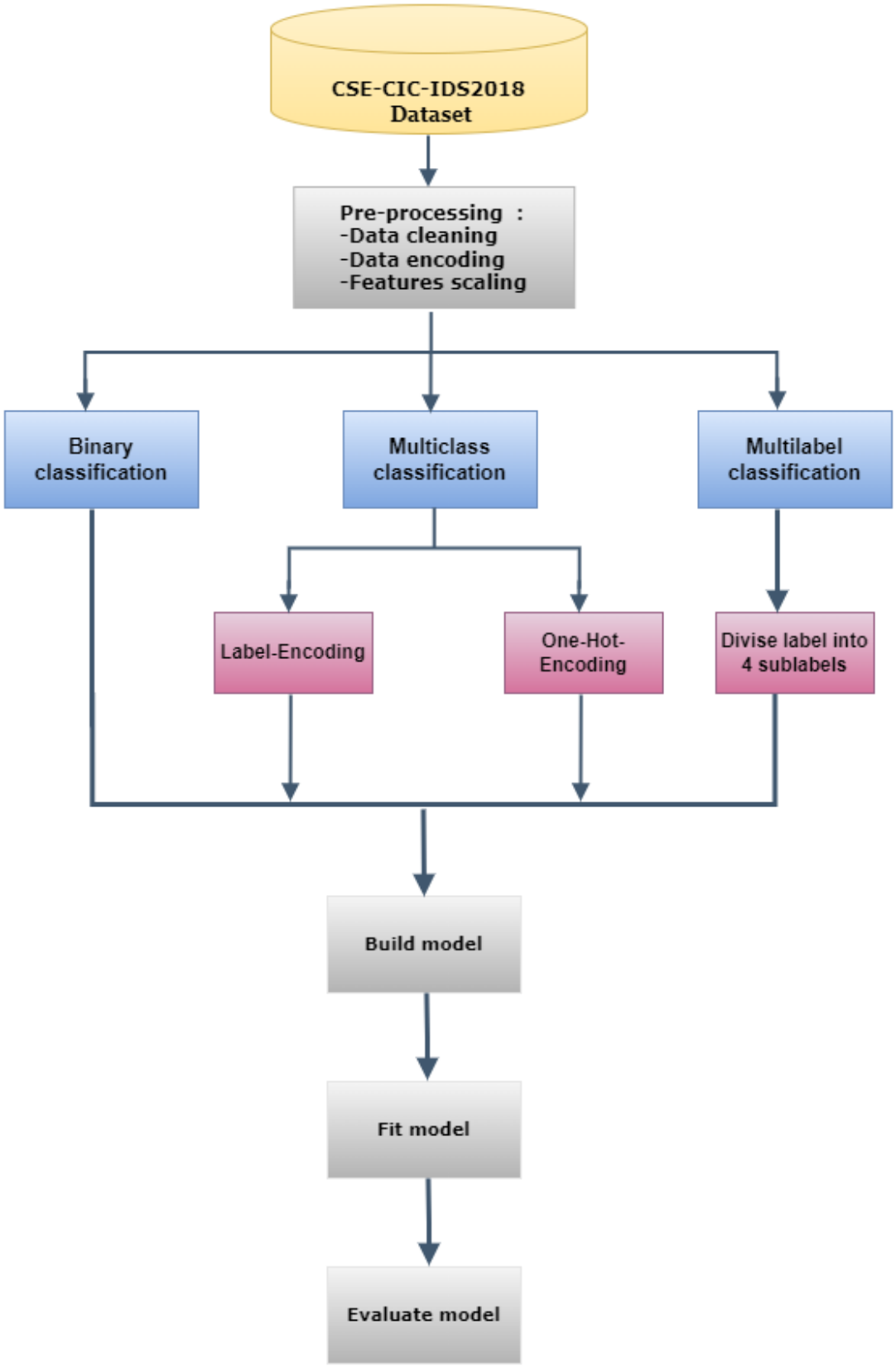


Figure 3.2 – *Composants architecturaux du modèle de réseau neuronal profond proposé.*

3.3.4 Classification multi-classes

Dans cette approche, nous avons utilisé deux méthodes pour coder la colonne ń Label ż, qui constitue notre cible.

3.3.4.1 Classification multi-classes - Label Encoding

L'encodage des étiquettes (Label Encoding) est une méthode simple et rapide qui consiste à transformer des variables catégorielles en variables numériques [122]. Cette technique remplace chaque catégorie par un entier unique. L'objectif de cette approche est de faciliter l'entraînement du modèle en rendant les données catégorielles compatibles avec les algorithmes d'apprentissage automatique et d'apprentissage profond.

Dans le cadre de cette étude, nous appliquons cette méthode à la variable cible du jeu de données, nommée "Label", qui contient quatre classes distinctes : Benign, DDoS attacks-LOIC-HTTP, DDoS attack-HOIC et DDoS attack-LOIC-UDP. Par l'application de la méthode d'encodage des étiquettes, ces catégories seront respectivement remplacées par les entiers 0, 1, 2 et 3 (voir Figure 3.3).

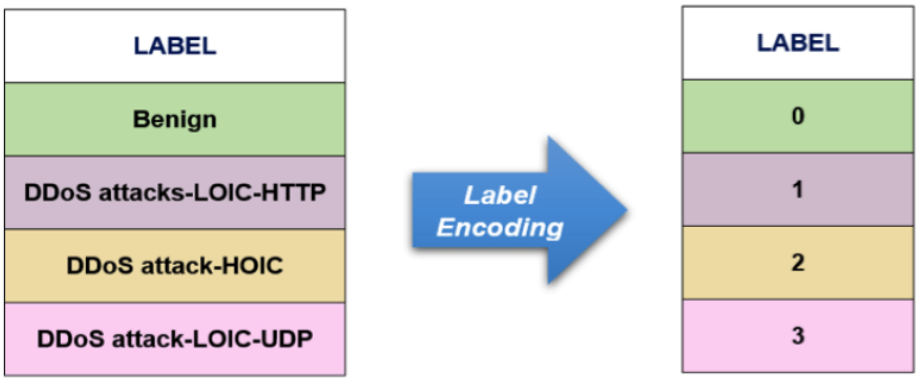


Figure 3.3 *– Label encoding.*

Les figures 3.4 et 3.5 montrent la perte et la précision du modèle pendant le processus d'apprentissage en fonction du nombre d'itérations.

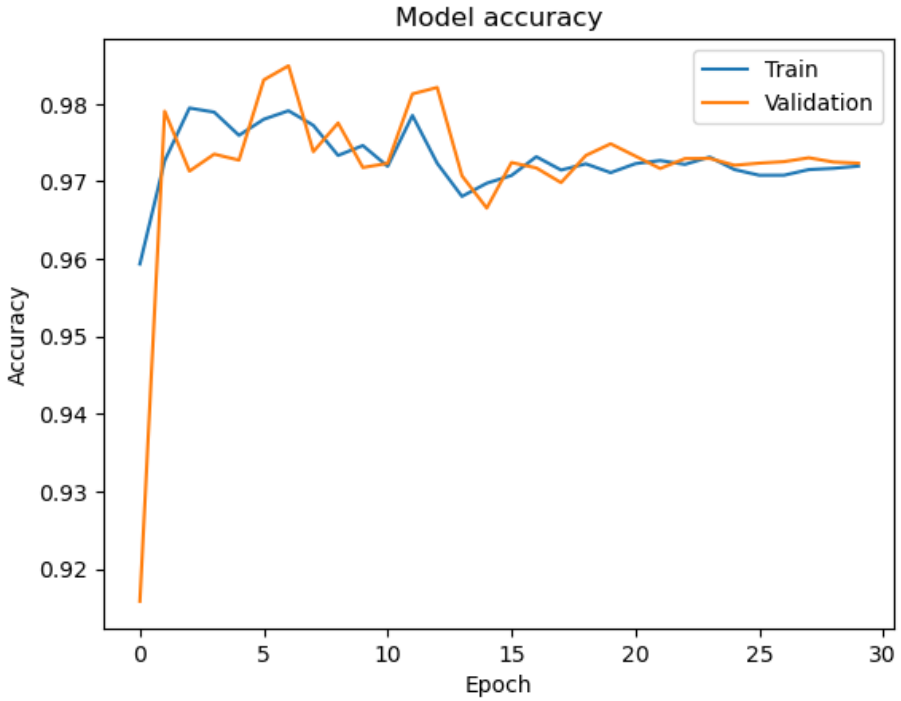


Figure 3.4 *– Training and Validation Accuracy - Multi-class (label encoding).*

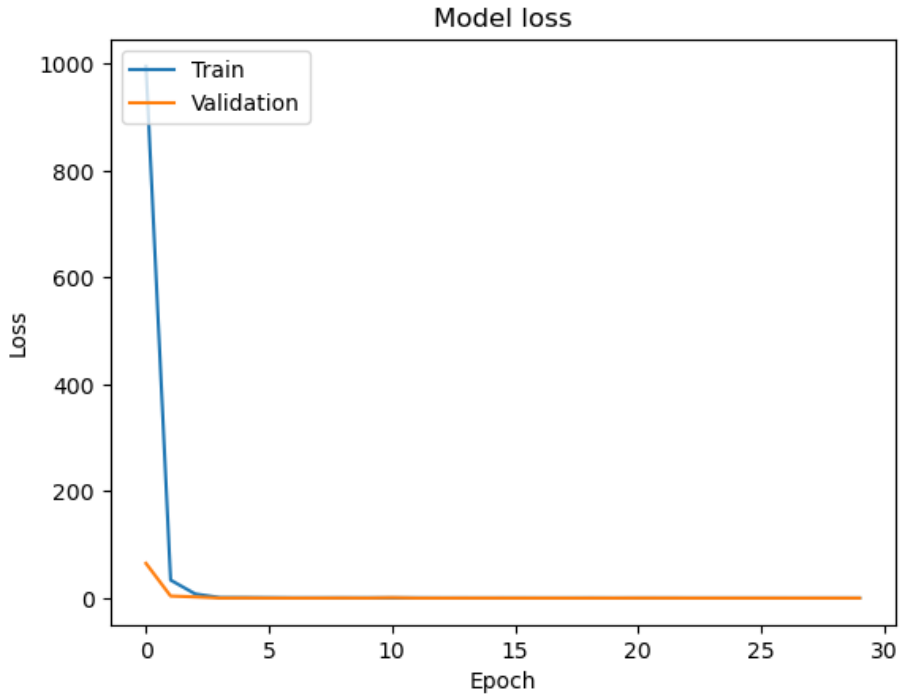


Figure 3.5 *– Training and Validation Loss - Multi-class (label encoding).*

En utilisant l'approche d'encodage des étiquettes (Label Encoding), le modèle a atteint une précision de test de 97,18 % et une précision de validation de 97,24 %. Toutefois, cette approche a conduit à une perte (Loss) plus faible, avec une valeur de

9,04 % sur les données de test et de 8,88 % sur les données de validation. La Figure 3.6 présente la matrice de confusion obtenue pour la classification multiclasses avec l'encodage des étiquettes.

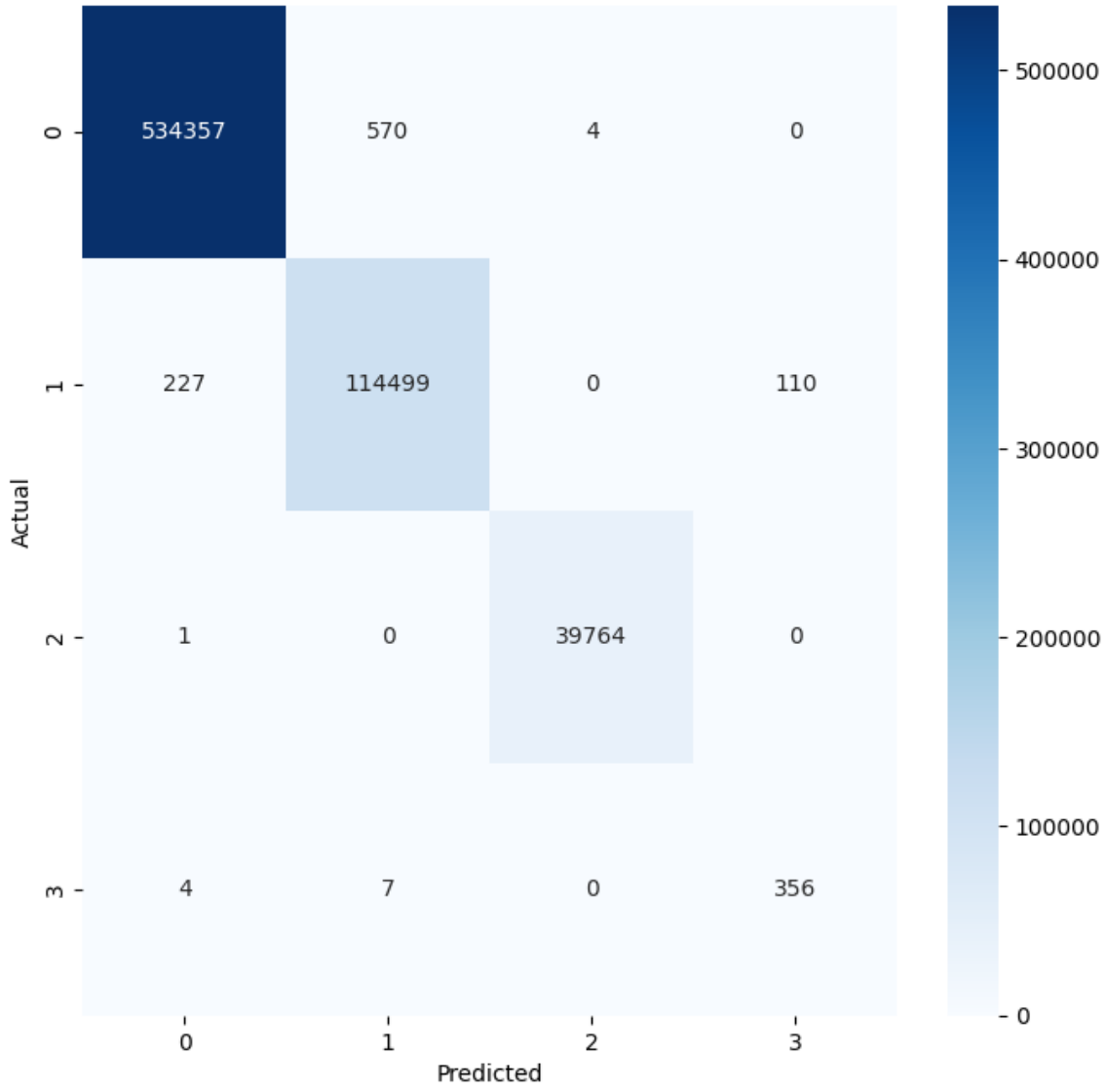


Figure 3.6 *– Matrice de confusion - Multi-class (label encoding).*

3.3.4.2 Classification multi-classes - One-hot encoding

Dans cette approche, nous utilisons une autre méthode appelée encodage one-hot (one-hot encoding) [123]. Cette méthode consiste à transformer chaque classe de la variable cible "Label" en un vecteur binaire unique. Chaque vecteur aura une longueur égale au nombre total de classes distinctes présentes dans la variable cible, et contiendra une seule valeur à 1, correspondant à la classe de l'observation considérée, tandis que les autres positions du vecteur prendront la valeur 0.

Par exemple, si l'observation appartient à la classe Benign, le vecteur binaire

associé sera : [1, 0, 0, 0] (voir Figure 3.7).

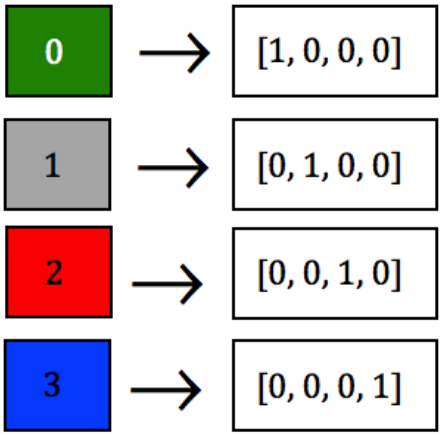


Figure 3.7 *– One-hot encoding.*

Les Figures 3.8 et 3.9 présentent respectivement l'évolution de la perte (Loss) et de la précision (Accuracy) du modèle au cours des itérations pendant le processus d'entraînement. Cette approche a permis d'obtenir des résultats très satisfaisants, avec une précision de 99,87 % sur les données de test et de 99,88 % sur les données de validation. Les résultats obtenus indiquent également des valeurs de perte (Loss) très faibles et stables, avec une perte de 0,45 % pour les données de test et de 0,44 % pour les données de validation.

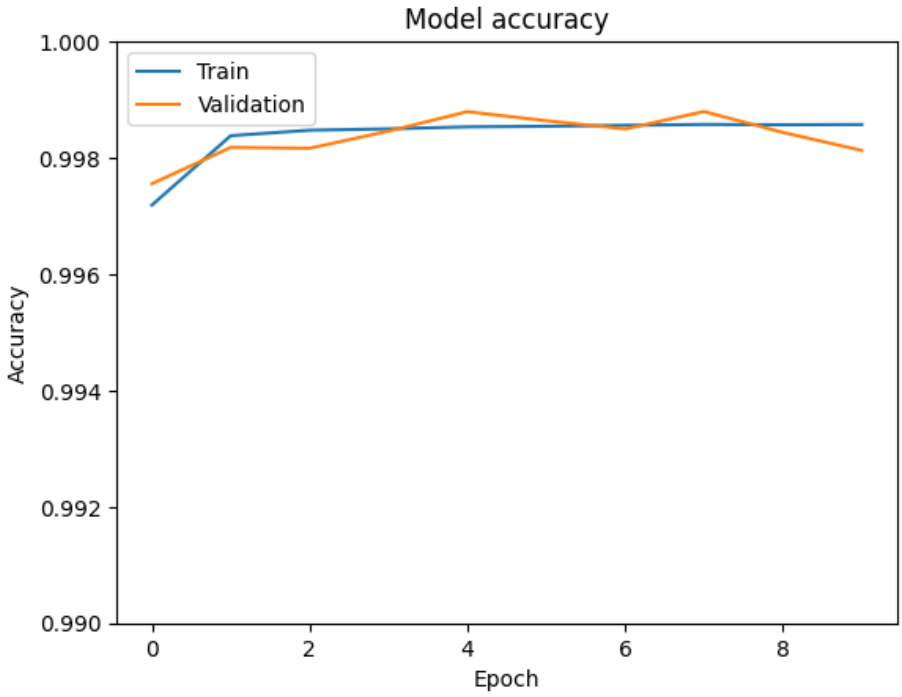


Figure 3.8 *– Training and Validation Accuracy - Multi-class (one-hot encoding).*

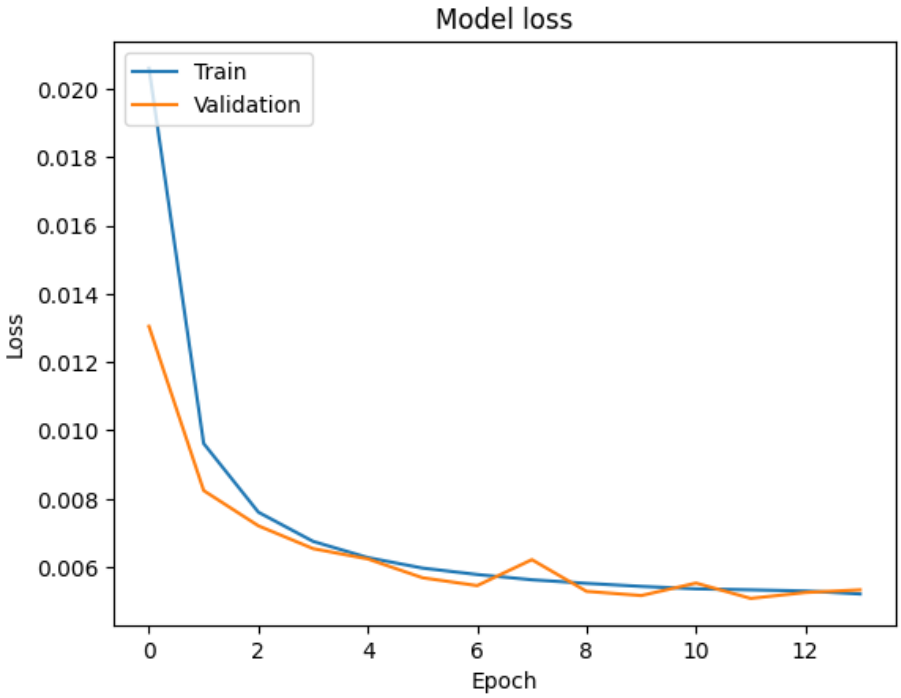


Figure 3.9 *– Training and Validation Loss - Multi-class (one-hot encoding).*

Les données présentées dans le Tableau 3.1 indiquent les métriques d'évaluation pour la classification multiclasses utilisant l'encodage one-hot. Ce tableau présente la précision globale (accuracy), la précision au sens strict (precision), le rappel (recall) ainsi que le score F1 (F1-score) pour chaque classe.

Le modèle présente une précision et une exactitude élevées pour l'ensemble des classes, à l'exception de la classe "DDoS attacks-LOIC-UDP". Le rappel atteint 100 % pour les classes Benign, DDoS attacks-LOIC-HTTP et DDoS attacks-HOIC, ce qui signifie que le modèle a correctement identifié l'ensemble des instances positives réelles pour chacune de ces classes. En revanche, le rappel pour la classe "DDoS attacks-LOIC-UDP" est de 98 %.

Cependant, la précision pour la classe "DDoS attacks-LOIC-UDP" n'est que de 74 %, ce qui indique que, parmi l'ensemble des instances classées comme positives pour cette étiquette, seulement 74 % correspondaient effectivement à des vrais positifs. Le score F1 pour cette classe est également inférieur aux autres, atteignant 85 %. La Figure 3.10 présente la matrice de confusion correspondante.

Tableau 3.1 *– Métriques d'évaluation pour la classification multiclasses (encodage one-hot).*

Label	**Accuracy**	**Precision**	**Recall**	**F1-score**
Benign	99.90%	100%	100%	100%
DDoS attacks-LOIC-HTTP	99.50%	99%	100%	100%
DDoS attack-HOIC	99.99%	100%	100%	100%
DDoS attacks-LOIC-UDP	98.28%	74%	98%	85%

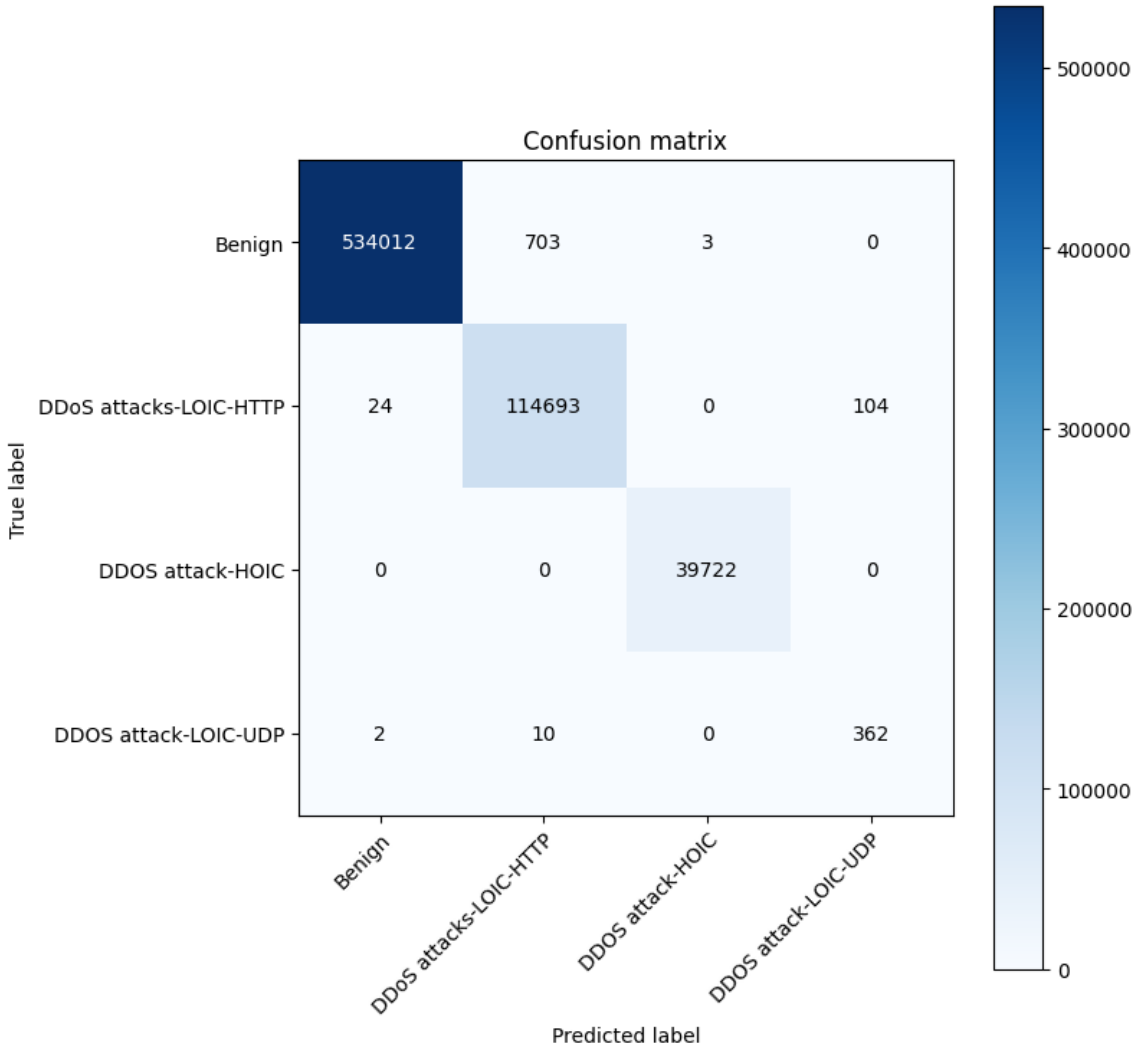


Figure 3.10 *– Matrice de confusion - Multi-class (one-hot encoding).*

3.3.5 Classification binaire

Dans cette approche, nous avons regroupé les trois types d'attaques DDoS, à savoir DDoS attacks-LOIC-HTTP, DDoS attack-HOIC et DDoS attack-LOIC-UDP,

en un seul type nommé ń DDoS ż, auquel s'ajoute le flux ń Benign ż. L'objectif de cette approche est de simplifier la variable cible en réduisant le nombre de classes, tout en maintenant une distinction importante entre les observations DDoS et Benign.

Les figures 3.11 et 3.12 présentent respectivement la "Loss" et la "Accuracy" du modèle en fonction des itérations durant le processus d'apprentissage. Cette approche a également produit des résultats très satisfaisants, avec une précision de 99,88 % sur les données de test et de 99,88 % sur les données de validation. Cette méthode a également généré des résultats satisfaisants pour la fonction de perte, avec une perte de 0,4 % sur les données de test et de 0,41 % sur les données de validation.

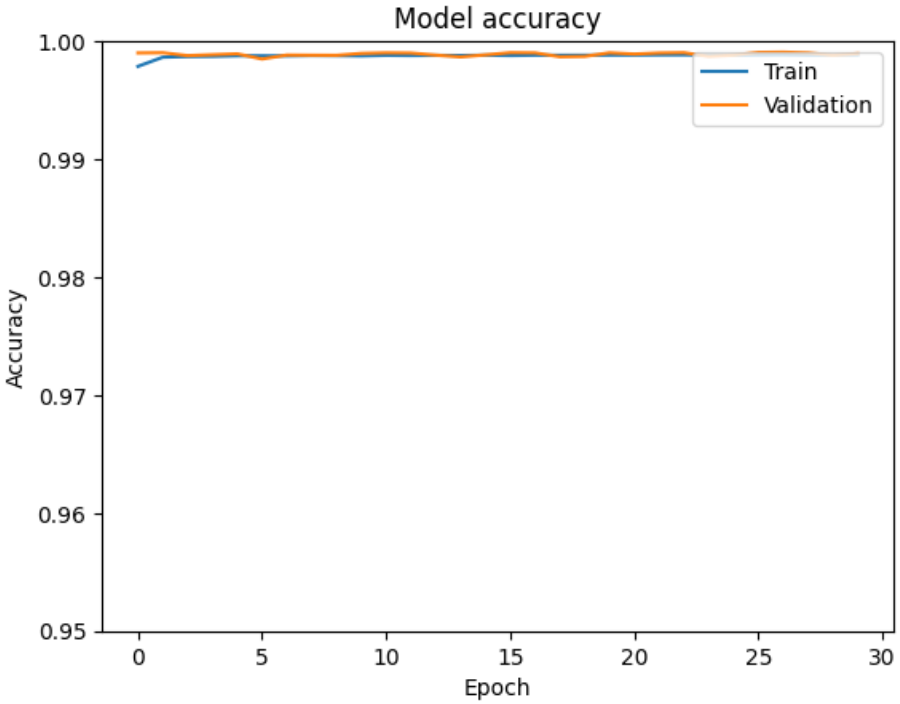


Figure 3.11 *– Training and Validation Accuracy - Classification binaire.*

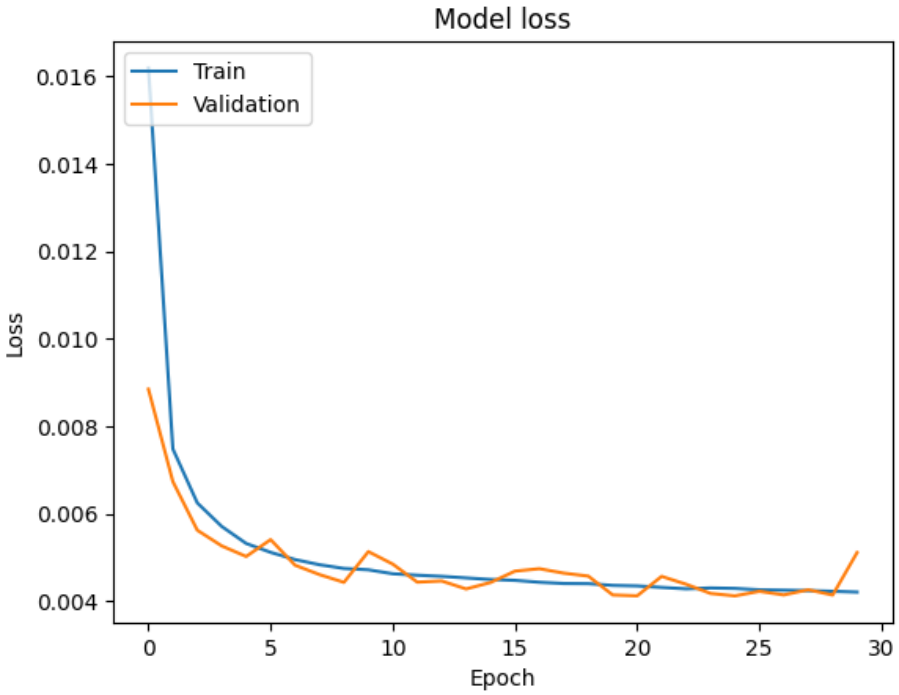


Figure 3.12 – *Training and Validation Loss - Classification binaire.*

Le tableau 3.2 présente les métriques d'évaluation pour la classification binaire en termes de précision globale (accuracy), de précision (precision), de rappel (recall) et de score F1 (F1-score) pour chaque étiquette. Pour les données considérées, le modèle a atteint une précision, une précision, un rappel et un score F1 parfaits pour les deux étiquettes, ń Benign ż et ń DDoS ż. Cela signifie que le modèle a correctement classé l'ensemble des instances pour les deux classes. La figure 3.13 présente la matrice de confusion correspondante.

Tableau 3.2 – *Métriques d'évaluation pour la classification binaire.*

Label	**Accuracy**	**Precision**	**Recall**	**F1-score**
Benign	100%	100%	100%	100%
DDoS	100%	100%	100%	100%

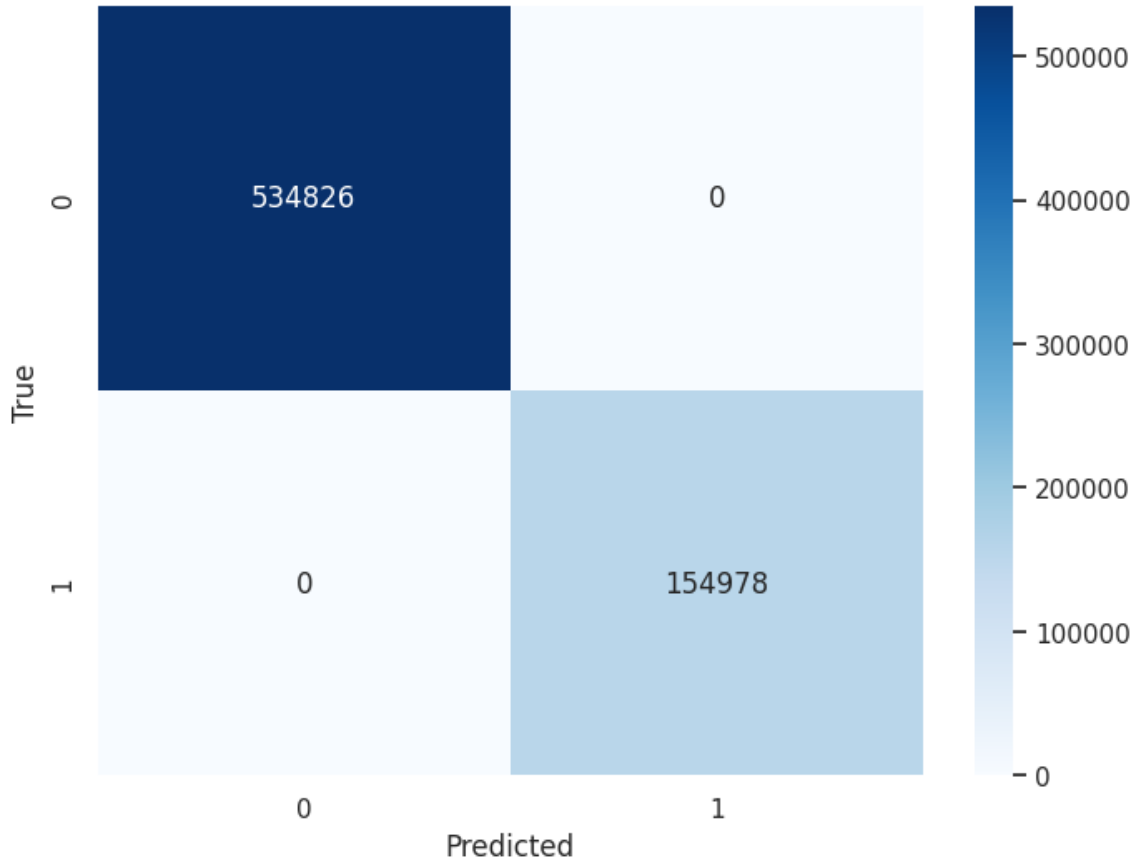


Figure 3.13 *– Matrice de confusion - Classification binaire.*

3.3.6 Classification multi-label

Dans cette approche, nous utilisons une technique de codage multi-étiquette (multi-label encoding) pour la variable cible. Cette méthode consiste à générer une nouvelle colonne binaire pour chaque catégorie distincte de la variable cible ń Label ż. Une valeur de 1 est attribuée à la colonne correspondant à la catégorie cible de l'instance, tandis qu'une valeur de 0 est attribuée à toutes les autres colonnes. Avec les quatre classes cibles Benign, DDoS-LOIC-HTTP, DDoS attack-HOIC et DDoS-HOIC-LOIC-UDP ce codage multi-étiquette génère ainsi quatre colonnes pour la variable cible, comme illustré à la figure 3.14. En transformant la variable cible catégorielle unique en plusieurs cibles binaires, ce codage permet une différenciation plus fine entre les classes.

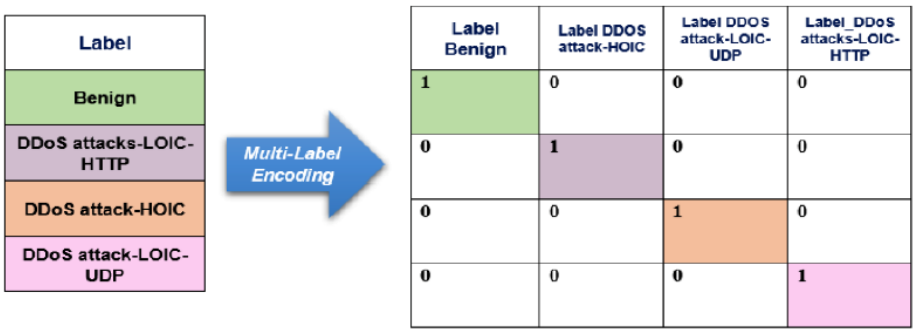


Figure 3.14 – *Multi-label encoding.*

L'objectif de cette approche était de permettre aux modèles d'opérer des distinctions plus fines entre les différentes catégories de la variable cible. Les figures 3.15 et 3.16 illustrent respectivement la fonction de perte (Loss) et la précision (Accuracy) du modèle au cours du processus d'apprentissage.

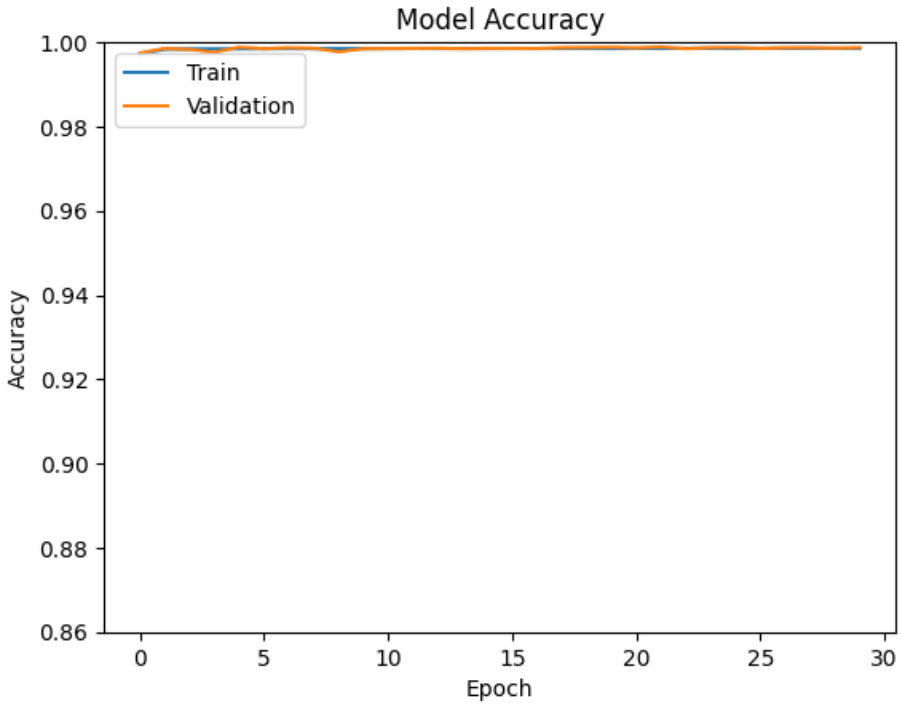


Figure 3.15 – *Training and Validation Accuracy - Multi-label encoding.*

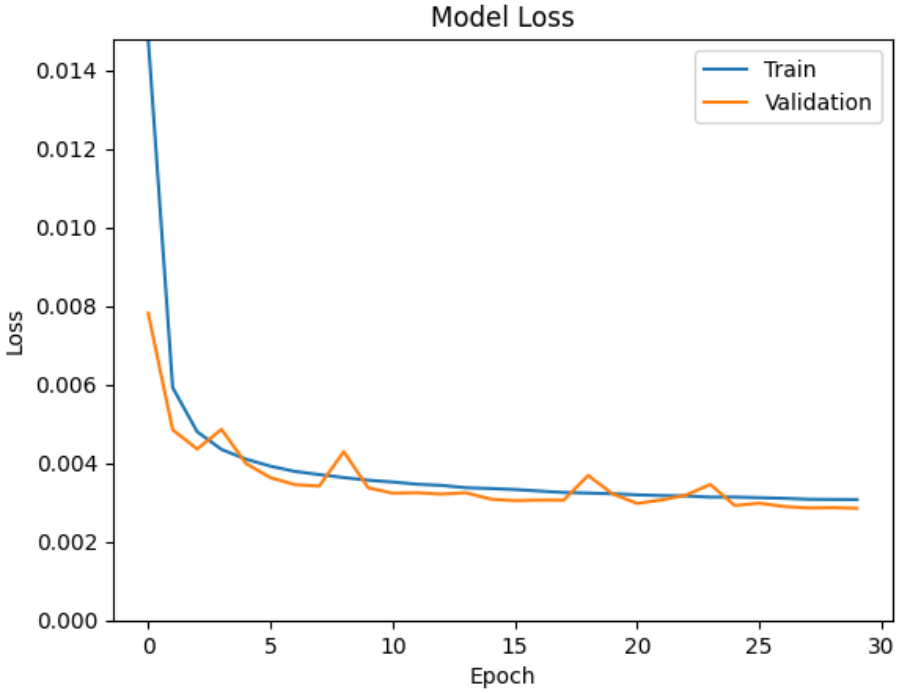


Figure 3.16 *– Training and Validation Loss - Multi-label encoding.*

La méthode de codage multi-étiquette a donné des résultats parfaits, avec une précision de 100 % à la fois sur les données de test et de validation. Cette approche a permis de transformer les variables catégorielles en une représentation à codage multiple, offrant ainsi une caractérisation plus riche des relations complexes entre les différentes catégories.

La méthode de codage multi-étiquette a également atteint des valeurs de perte très faibles : 8,15 % sur les données de test et 11,8 % sur les données de validation. Ces faibles pertes indiquent que le modèle a réussi à apprendre à classer les données avec un surapprentissage minimal. Globalement, les faibles pertes observées sur les données de test et de validation démontrent l'efficacité de l'approche de codage multi-étiquette pour ce problème.

Le tableau 3.3 présente les métriques d'évaluation pour les 4 étiquettes (0, 1, 2, 3) dans le cadre de la classification multi-étiquette. L'étiquette 0 affiche une très haute précision globale (99,89 %), une précision (100 %), un rappel (100 %) et un score F1 (100 %). Cela indique que le modèle prédit correctement l'étiquette 0 lorsqu'elle est présente et ne la prédit pas lorsqu'elle est absente. L'étiquette 1 présente également une précision globale élevée (99,70 %) et un rappel parfait (100 %), mais une précision légèrement inférieure (99 %). Cela signifie que le modèle prédit très correctement l'étiquette 1 lorsqu'elle est présente, mais peut occasionnellement la prédire lorsqu'elle est absente. Le score F1 permet de pondérer la précision et le rappel. L'étiquette 2 obtient des scores parfaits pour l'ensemble des métriques, indiquant que le modèle prédit parfaitement cette étiquette. L'étiquette 3 présente les scores les plus faibles :

précision globale (97 %), précision (76 %), rappel (97 %) et score F1 (86 %). Le modèle rencontre plus de difficultés avec cette étiquette par rapport aux autres. La faible précision indique qu'il prédit fréquemment l'étiquette 3 lorsqu'elle est absente. Globalement, le modèle offre de très bonnes performances sur les étiquettes 0, 1 et 2, avec des niveaux élevés de précision, de rappel et de score F1. Il rencontre davantage de difficultés avec l'étiquette 3, pour laquelle une amélioration de la précision pourrait encore optimiser les performances globales. La figure 3.17 présente la matrice de confusion correspondante.

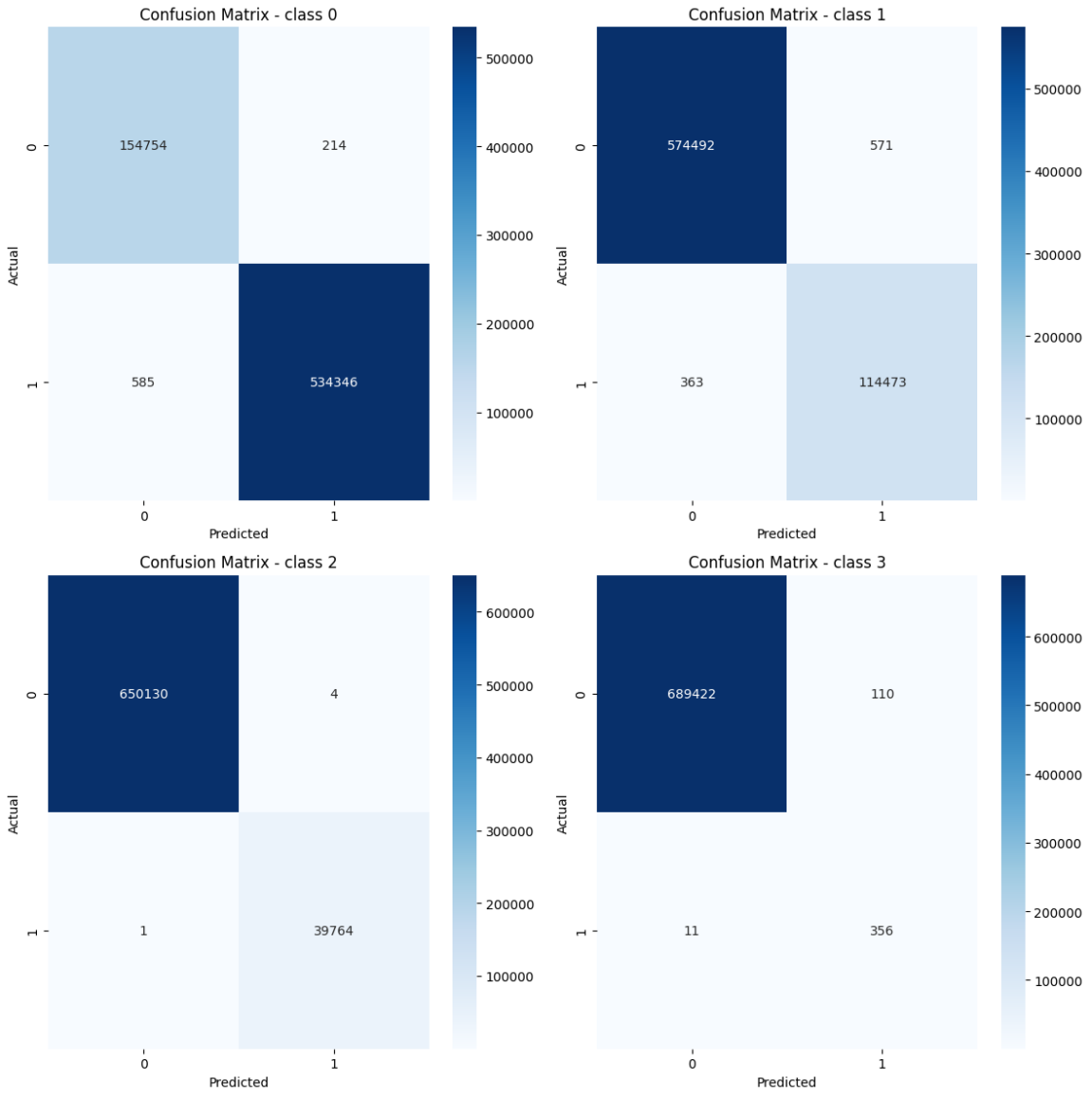


Figure 3.17 – *Matrice de confusion - Classification multi-label.*

Tableau 3.3 *– Métriques d'évaluation pour la classification multi-label.*

Label	**Accuracy**	**Precision**	**Recall**	**F1-score**
0	99.89%	100%	100%	100%
1	99.70%	99%	100%	100%
2	99.99%	100%	100%	100%
3	97%	76%	97%	86%

En comparant les différentes techniques de codage, les approches de classification binaire et de codage multi-classe en one-hot encoding ont produit des résultats comparables et très efficaces, avec des métriques similaires en termes de précision et de perte. La méthode de codage par label encoder a présenté des performances légèrement inférieures selon ces critères d'évaluation. Il convient de noter que l'approche de codage multi-étiquette a atteint une précision parfaite et une perte remarquablement faible, bien que ces résultats idéaux puissent être spécifiques au jeu de données utilisé et aux objectifs de l'analyse menée. Globalement, ces différences soulignent l'importance de sélectionner la stratégie de codage la plus adaptée au contexte du problème, aux caractéristiques des données et aux objectifs du travail de modélisation.

Pour l'ensemble des techniques de codage, les modèles ont généré des valeurs de perte faibles, traduisant une réduction efficace de l'erreur de prédiction. Toutefois, la méthode label encoder a présenté des pertes plus élevées, confirmant son moindre niveau d'adéquation à ce jeu de données par rapport aux autres approches. En revanche, les codages one-hot multi-classe, binaire et multi-étiquette ont conduit à des pertes faibles et comparables, soulignant leur efficacité dans le cadre de ce problème de classification. La similitude des faibles pertes obtenues avec ces trois méthodes atteste de leur aptitude équivalente à modéliser les relations et les structures présentes dans les données. Bien que le label encoder se soit avéré ici moins optimal, les autres stratégies ont permis de minimiser efficacement l'erreur et la perte, validant ainsi leur pertinence pour cette tâche d'apprentissage.

Enfin, le tableau 3.4 présente les résultats des différentes approches proposées pour la classification des données, en comparant les métriques de précision globale (accuracy), de précision (precision), de rappel (recall) et de score F1 (F1-score) pour chaque méthode. La classification binaire a obtenu des scores parfaits avec une précision, une précision, un rappel et un score F1 de 100 %, indiquant que le modèle distingue parfaitement les deux classes. Le codage one-hot présente une précision très élevée (99,84 %), mais une précision (93,25 %) et un score F1 (96,25 %) légèrement inférieurs. Le rappel presque parfait (99,5 %) montre que le modèle omet rarement un cas positif, tandis que la précision plus faible indique la présence de quelques faux positifs. Le codage multi-étiquette affiche également une précision élevée (99,87 %), avec

une précision (93,75 %), un rappel (99,25 %) et un score F1 (96,5 %) similaires. Cela indique également très peu de faux négatifs mais quelques faux positifs. Globalement, le classificateur binaire présente des performances parfaites. Les modèles multi-classes présentent également une grande précision avec de rares erreurs de classification.

Tableau 3.4 *– Métriques globales pour les approches proposées.*

Approche	**Accuracy**	**Precision**	**Recall**	**F1-score**
Binaire	99.89%	100%	100%	100%
One-hot encoding	99.84%	93.25%	99.5%	96.25%
Multi-label	99.87%	93.75%	99.25%	96.5%

3.4 Détection d'intrusion basée sur les réseaux neuronaux profonds à l'aide du dataset Edge-IIoTSet

3.4.1 Dataset

Edge-IIoTset est un jeu de données réaliste en cybersécurité destiné aux applications IoT et IIoT, pouvant être utilisé pour l'entraînement des systèmes de détection d'intrusion à l'aide d'approches d'apprentissage automatique centralisées ou fédérées. Ce jeu de données est structuré en sept couches : informatique en nuage (cloud computing), virtualisation des fonctions réseau (network functions virtualization), blockchain, informatique en brouillard (fog computing), réseaux définis par logiciel (software-defined networking), informatique en périphérie (edge computing) et couche de perception IoT/IIoT. Chaque couche s'appuie sur des technologies émergentes telles que les contrôleurs ONOS SDN, les courtiers Mosquitto (Mosquitto brokers sont des courtiers de messages open source qui implémentent le protocole MQTT pour la communication légère entre appareils, en particulier dans le contexte de l'Internet des Objets), OPNFV, Hyperledger Sawtooth, ThingsBoard, les jumeaux numériques (digital twins) et Modbus TCP/IP, afin de répondre aux principales exigences des environnements IoT/IIoT. Les données IoT proviennent de plus de dix types de dispositifs différents, incluant des capteurs à faible coût pour l'humidité, la température, le pH, l'humidité du sol, les ultrasons, le niveau d'eau, la détection de flamme, la fréquence cardiaque, entre autres. Le jeu de données couvre 14 attaques liées aux protocoles de connectivité IoT/IIoT, classées en cinq catégories de menaces : déni de service/déni de service distribué (DoS/DDoS), collecte d'informations (information gathering), attaque de type homme du milieu (man-in-the-middle), injection et logiciels malveillants (malware) [124].

Lors du chargement initial du jeu de données, l'objectif est d'identifier une va-

riable cible à prédire et les variables explicatives à utiliser pour modéliser cette cible. Un prétraitement est nécessaire afin d'éliminer les valeurs manquantes et les doublons, d'encoder les colonnes catégorielles, ainsi que de mettre à l'échelle les variables numériques par des techniques de normalisation ou de standardisation. Étant donné que le jeu de données Edge-IIoTset contient un volume important de données, la séparation des ensembles d'entraînement et de test est réalisée selon un ratio de 98 % pour l'entraînement et 2 % pour le test.

Dans ce travail, l'objectif est de détecter et de classifier les attaques en trois groupes : ń Normal ż, ń DDoS Attacks ż et ń Different Attacks ż étiqueté ń DA ż. Le groupe DA regroupe les attaques suivantes : Injection_Attack, Malware-Attack, Scanning-Attack et MITM.

Une représentation visuelle de la variable cible est présentée sous forme de graphique de tendance, comme illustré à la figure 3.18. La hauteur des barres représente le trafic ń Normal ż, tandis que les deux autres barres représentent respectivement les ń DDoS Attacks ż et les attaques ń DA ż. Chacune des valeurs textuelles de la variable cible a été convertie en valeur numérique : Normal est remplacé par 0, DDoS Attack par 1, et DA par 2.

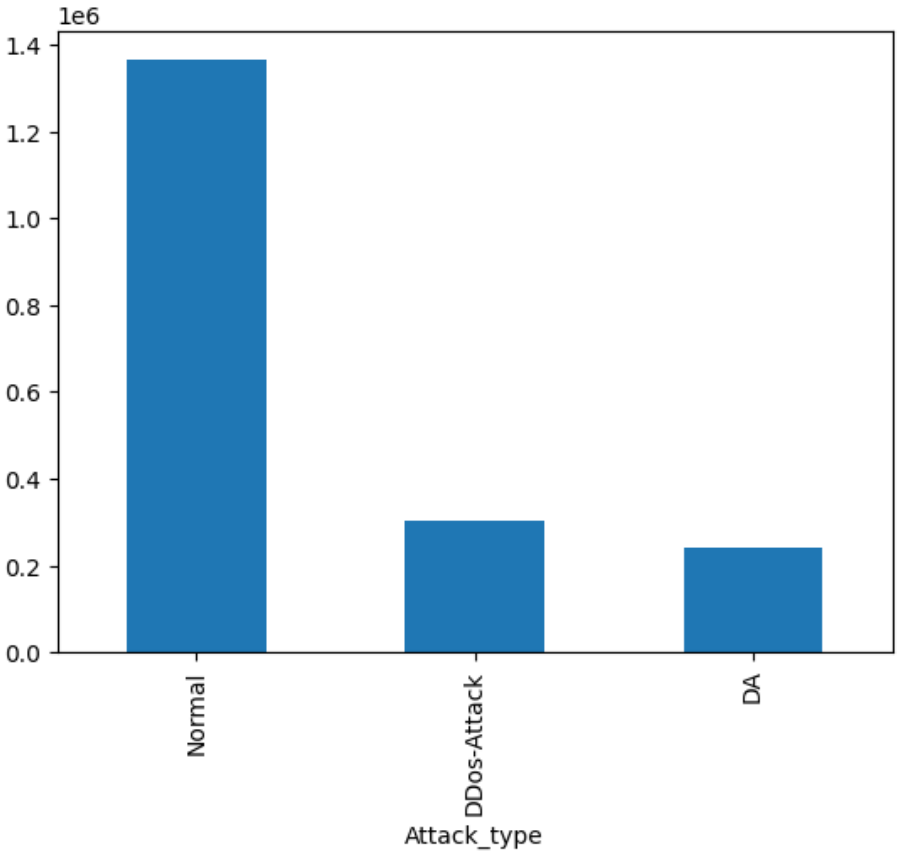


Figure 3.18 – *Graphique en barres de la distribution des valeurs de la variable cible.*

3.4.2 Architecture du modèle

Le modèle construit est défini comme une séquence de couches. Nous avons créé un modèle Sequential et ajouté cinq couches cachées. La première couche cachée comporte 256 unités et attend 96 variables d'entrée (caractéristiques). Les deuxième, troisième, quatrième et cinquième couches cachées comportent respectivement 128, 64, 32 et 16 unités. Enfin, la couche de sortie contient 3 unités permettant de détecter les trois types d'attaques. Nous utilisons la fonction d'activation rectifier (ReLU) pour l'ensemble des couches cachées, et la fonction d'activation softmax pour la couche de sortie.

Nous appliquons la technique de régularisation L2 avec un paramètre de régularisation (lambda) égal à 0,00001 sur les cinq couches cachées, afin d'éviter le surapprentissage du modèle sur les données d'entraînement. Le processus d'apprentissage est exécuté sur un nombre fixe d'itérations, égal à 200 époques (epochs). Le paramètre batch_size est fixé à 1000, étant donné le volume important de données du jeu de données. La métrique utilisée pour évaluer les performances du modèle est la précision (accuracy). Dans ce modèle, la fonction de perte utilisée est l'entropie croisée catégorielle (categorical cross-entropy). L'optimiseur Adam (Adaptive Moment Estimation) est utilisé pour calculer les poids optimaux des couches du classificateur, avec un taux d'apprentissage fixé à 0,0001.

Figure 3.19 *– A plot of training and validation accuracies.*

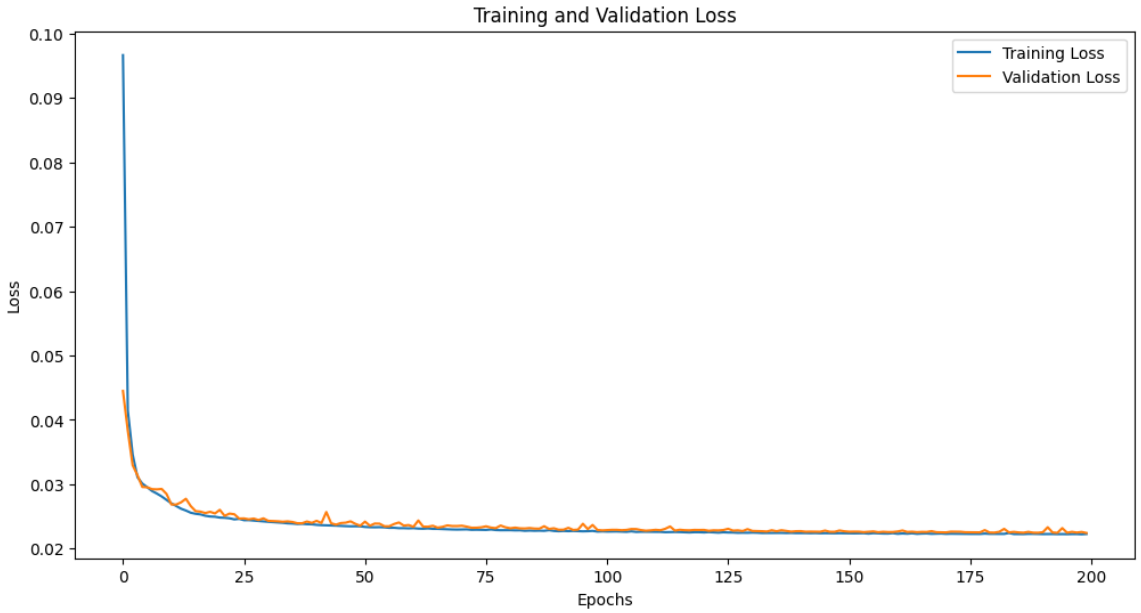


Figure 3.20 *– A plot of training and validation losses.*

Dans les expériences menées, la valeur du nombre d'époques a été fixée à 200 lors de l'apprentissage du modèle. Les performances obtenues avec le jeu de données Edge-IIoTset ont atteint des résultats proches de 100 %, comme l'illustre la figure 3.19. Le taux d'erreur a été observé à une valeur de 0,022, proche de zéro, indiquant que la méthode de régularisation L2 appliquée a permis d'éviter le surapprentissage du modèle sur les données d'entraînement (voir figure 3.20). Globalement, le modèle a atteint une précision de 99,05 %.

Le tableau 3.5 présente la précision de classification pour les trois classes : Normal, DDoS-attacks et Different attacks. Pour la classe Normal, la précision est de 0,99996, ce qui est extrêmement élevé. Cela signifie que le modèle est presque parfaitement capable d'identifier le trafic réseau normal. Pour la classe DDoS-attacks, la précision atteint 0,999498, indiquant également une très haute performance du modèle dans la détection des attaques DDoS. En revanche, la précision pour la classe Different attacks est plus faible, à 0,91803. Bien qu'elle reste relativement élevée, cette valeur suggère que le modèle est moins efficace pour détecter les autres types d'attaques qui ne sont pas de type DDoS. En résumé, le modèle présente d'excellentes performances pour les classes Normal et DDoS-attacks, avec des précisions supérieures à 99 % pour les deux. Toutefois, il rencontre davantage de difficultés pour les autres types d'attaques, n'atteignant une accuracy d'environ 91 %.

Tableau 3.5 *– Précision de classification pour chaque classe*

Classe	**Précision (Accuracy)**
Normal	0.9999632960176179
DDoS-attacks	0.9994982438534872
Different attacks	0.9180327868852459

Le rapport de classification pour le modèle prédisant trois classes (0, 1, 2) est présenté dans le tableau 3.6. Pour la classe 0, les scores de précision et de rappel sont parfaits avec des valeurs de 1,0, ce qui signifie que le modèle prédit parfaitement cette classe sans aucun faux positif ni faux négatif. Le score F1 est également parfait à 1,0.

Pour la classe 1, la précision est de 0,94 et le rappel de 1,0. Le rappel élevé indique que le modèle prédit correctement tous les exemples de cette classe, mais la précision légèrement inférieure à 0,94 révèle la présence de quelques faux positifs. Le score F1, qui équilibre ces deux métriques, est de 0,97.

Pour la classe 2, la précision est parfaite à 1,0, mais le rappel est plus faible à 0,92. Cela signifie que le modèle ne parvient pas à prédire correctement tous les exemples de cette classe, bien que toutes les prédictions effectuées soient correctes (absence de faux positifs). Le score F1 atteint 0,96.

Globalement, le modèle présente de très bonnes performances, avec des scores F1 supérieurs à 0,90 pour l'ensemble des classes. La classe 0 semble être la plus facile à prédire parfaitement pour le modèle, tandis que la classe 2 reste plus difficile en raison de son rappel plus faible.

Tableau 3.6 *– Rapport de classification*

Classe	**Précision (Precision)**	**Rappel (Recall)**	**Score F1 (F1-score)**
0	100%	100%	100%
1	94%	100%	97%
2	100%	92%	96%

La matrice de confusion du modèle est donnée par la figure 3.21.

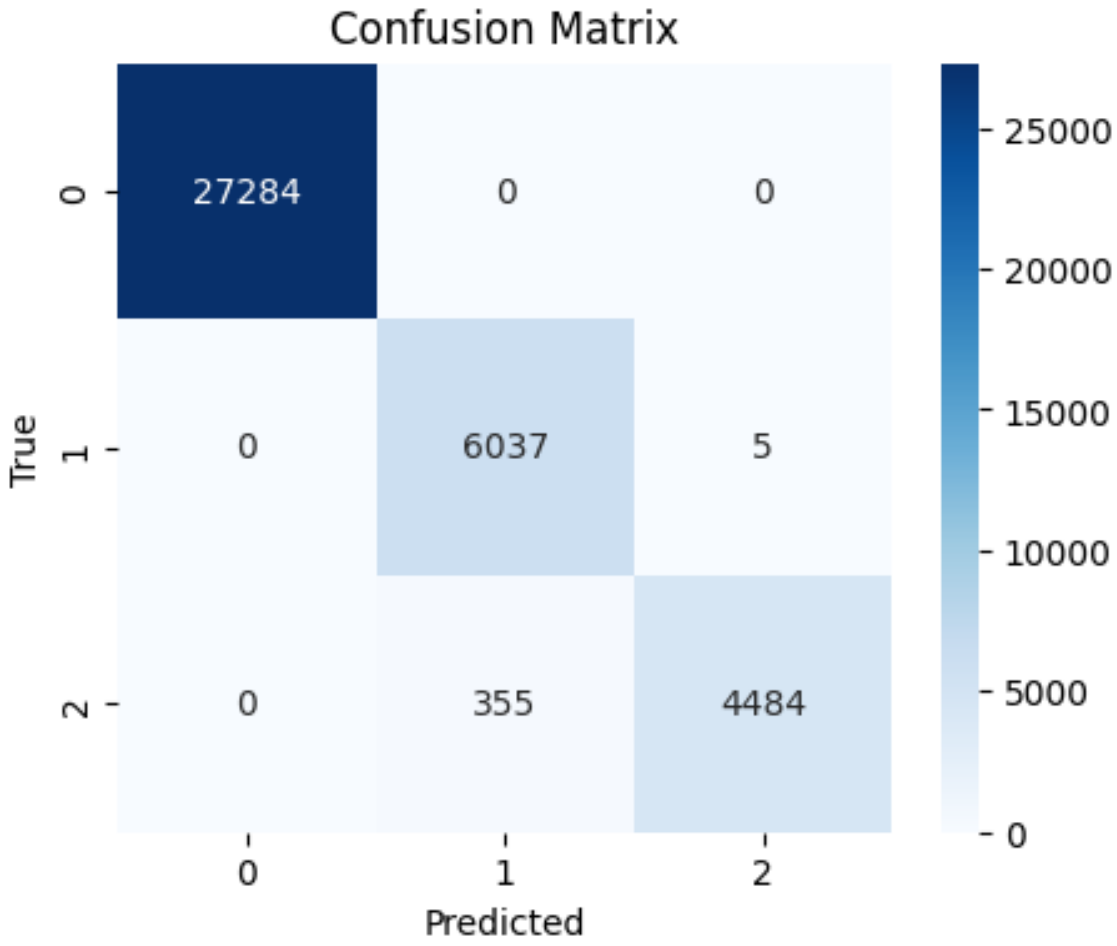


Figure 3.21 *– Matrice de confusion.*

3.5 Etude comparative

Dans cette section, une analyse comparative numérique est présentée sur la base de différentes études discutées dans la revue de littérature, afin de vérifier l'efficacité des modèles proposés.

Le tableau 3.7 compare les performances (en termes de précision) des modèles proposés avec les travaux de recherche antérieurs pour la détection des attaques DDoS sur différents jeux de données. Le modèle DNN proposé atteint une précision de 99,05 % sur le jeu de données EdgeIoTSet, surpassant le meilleur résultat précédent de 98,8 % obtenu par Alashhab et al.. Sur le jeu de données CSE-CIC-IDS2018, le modèle DNN proposé atteint également la précision la plus élevée avec 99,87 %. Les autres méthodes répertoriées atteignent des précisions variant de 93 % à 96 % sur les autres jeux de données (NSL-KDD, CAIDA, CIC et un jeu de données personnalisé).

Le tableau montre que les deux modèles DNN proposés surpassent les méthodes existantes, obtenant les meilleurs résultats respectivement sur les jeux de données EdgeIoTSet et CSE-CIC-IDS2018.

Tableau 3.7 *– Comparaison des performances des méthodes proposées avec les contributions de la littérature*

Auteur	**Dataset**	**Accuracy**
Zhijun et al. [110]	NSL-KDD	95.8%
Yang et al. [111]	Personnalisé	96%
Sudar & Deepalakshmi [112]	CIC	93%
Alashhab et al. [113]	Edge-IIoTset	98.8%
Proposed DNN Models	Edge-IIoTset, CSE-CIC-IDS2018	99.05%, 99.87%

3.6 Conclusion

Le travail présenté dans ce chapitre porte sur la conception et le développement de deux modèles d'apprentissage profond destinés à la détection et à la classification des attaques DDoS, en s'appuyant sur les jeux de données CSE-CIC-IDS2018 et Edge-IIoTset. Un prétraitement des données a été appliqué afin d'éliminer les valeurs NaN, les valeurs infinies, les lignes dupliquées ainsi que les colonnes non pertinentes susceptibles de dégrader les performances du modèle.

Par ailleurs, nous avons constaté que la structure et la qualité des données utilisées pour l'entraînement des modèles jouent un rôle déterminant dans les performances obtenues. Nous avons également étudié l'impact de différents hyperparamètres sur les performances des modèles d'apprentissage profond, tels que le nombre de couches cachées, le nombre de neurones par couche, les fonctions d'activation, l'optimiseur, le type et la valeur de la régularisation, le nombre d'itérations, le taux d'apprentissage, etc., qui influencent de manière significative la qualité des résultats.

Dans notre étude, et dans le but de détecter les attaques DDoS, nous avons adopté des approches de classification binaire, multi-classe et multi-étiquette sur le jeu de données CSE-CIC-IDS2018, et uniquement la classification multi-classe sur le jeu de données Edge-IIoTset. Ces différentes approches nous ont permis d'évaluer les performances de chaque modèle dans la détection des attaques DDoS.

Les résultats ont montré que la classification binaire a obtenu la meilleure précision, suivie par la classification multi-étiquette, puis la classification multi-classe lors de l'utilisation du jeu de données CSE-CIC-IDS2018.

Chapitre 4

Détection d'Anomalies pour les Systèmes de Prévention d'Intrusion basée sur l'Apprentissage Profond (CSE-CIC-IDS2018)

Sommaire

4.1 Introduction

Au cours des dernières décennies, les systèmes d'information ont transformé de nombreux domaines, tels que les réseaux informatiques, l'Internet, le World Wide Web (WWW) et l'Internet des Objets (IoT). Cette progression a entraîné une augmentation considérable des données statiques et dynamiques, accroissant ainsi les risques de menaces pour l'infrastructure mondiale de l'information. Selon le Computer Emergency Response Team (CERT) de l'Université Carnegie Mellon (CMU), les vulnérabilités aux attaques ont augmenté de manière exponentielle au cours de la dernière année [125].

Les cyberattaques englobent un large éventail de menaces pour les systèmes informatiques, allant des logiciels malveillants comme les virus aux accès non autorisés via le piratage et le craquage de mots de passe. Ces attaques sont devenues de plus en plus sophistiquées, ce qui a motivé le développement de techniques avancées de sécurité informatique au cours de la dernière décennie.

Un système de sécurité robuste protège les informations cruciales d'une entreprise contre les attaques potentielles. Les politiques de confidentialité varient selon les entreprises et les organisations en raison de la diversité de leurs missions. Garantir la confidentialité, l'intégrité et la disponibilité des sources d'information constitue un défi fondamental dans la conception des systèmes de sécurité.

La protection des systèmes d'information et des réseaux est généralement assurée en limitant l'accès aux ressources système via des méthodes telles que les pare-feux, les logiciels antivirus, le chiffrement/déchiffrement, le cryptage des messages, les mécanismes de protection par mot de passe et les protocoles réseau sécurisés.

Bien que les mesures de sécurité traditionnelles jouent un rôle essentiel, la nature dynamique des données sur Internet pose des défis majeurs. L'évolution constante des méthodes d'attaque nécessite une approche plus proactive. L'extraction de connaissances à partir des données de trafic réseau apparaît comme une solution clé. En analysant l'activité du réseau, les systèmes de sécurité peuvent identifier les menaces potentielles en temps réel. Cette approche conduit au développement des systèmes de détection d'intrusion (IDS).

Cependant, la création d'IDS efficaces présente des défis. Les classificateurs doivent être hautement adaptatifs pour reconnaître les nouvelles formes d'attaques et les distinguer des activités réseau légitimes. Cela nécessite un raffinement et une adaptation continus pour garantir que le système reste efficace face à un paysage de menaces en constante évolution. Cette dynamique souligne l'importance de la recherche et du développement continus dans les technologies de détection d'intrusion [126].

Le terme "intrusion" désigne tout accès non autorisé par un utilisateur compromettant la sécurité, l'intégrité, la disponibilité et la confidentialité des ressources connectées au réseau [127].

La confidentialité garantit que les ressources ne sont accessibles qu'aux utilisa-

teurs autorisés, généralement assurée par des techniques cryptographiques. L'intégrité garantit que les données ne sont pas altérées pendant leur transmission, avec des signatures numériques servant à vérifier leur authenticité [128].

Les systèmes de détection d'intrusion (IDS) sont conçus pour détecter les intrusions rapidement et avec précision [129], [130]. Intégrés aux systèmes informatiques, ils identifient les menaces en supposant que les vulnérabilités sont exploitées via une utilisation anormale. Tous les IDS s'appuient sur cette théorie à divers degrés [131].

Les méthodes de détection d'intrusion se divisent en trois catégories principales :

1. Détection basée sur les signatures (SD) : Identifie les attaques connues en comparant des motifs ou chaînes avec les événements capturés. Elle repose sur une connaissance préalable des attaques (détection "knowledge-based" ou "misuse").
2. Détection basée sur les anomalies (AD) : Détecte les écarts par rapport aux normes établies en surveillant les activités habituelles. Elle utilise des profils de comportement normal (statiques ou dynamiques) pour repérer des attaques critiques (ex : tentatives d'intrusion, spoofing, DoS). Aussi appelée détection "behavior-based".
3. Analyse protocolaire avec état (SPA) : Suit l'état des protocoles réseau pour vérifier la conformité des requêtes/réponses. Basée sur des profils génériques standardisés (ex : IETF), elle est aussi appelée détection "specification-based" [132].

Certains systèmes utilisent des approches d'apprentissage automatique (ML) et d'apprentissage profond (DL), incluant les réseaux de neurones (NN) [133]. Le DL, notamment via les réseaux neuronaux convolutifs (CNN) et récurrents (RNN), a révolutionné la détection d'intrusion en apprenant des motifs complexes à partir de vastes jeux de données. Contrairement aux méthodes traditionnelles, ces modèles identifient automatiquement les menaces sans règles prédéfinies.

Les IDS basés sur le DL ont montré des résultats impressionnants pour détecter diverses attaques (ex : déni de service, malware, intrusions réseau). Leur force majeure réside dans leur adaptabilité, apprenant à partir de données historiques pour identifier des attaques émergentesun atout crucial face aux menaces dynamiques où les méthodes classiques échouent.

Si un mode de vie entièrement connecté offre des avantages, la conscience des cybermenaces reste essentielle. Les IDS et modèles de DL améliorent significativement notre capacité à nous en protéger. Ce chapitre étudie les méthodes de DL pour la détection d'intrusion, mettant en lumière les approches les plus efficaces et fiables.

4.2 Travaux connexes

Le domaine des systèmes de détection d'intrusion (IDS) connaît une vague d'innovation grâce aux progrès de l'apprentissage automatique (ML) et de l'apprentissage

profond (DL). Ces algorithmes puissants, couplés à la disponibilité de vastes jeux de données, permettent le développement d'IDS plus efficaces et sophistiqués. Cette section présente un aperçu des travaux récents liés aux IDS.

4.2.1 Taxonomies et Revues des Méthodes IDS

Dans [134], les auteurs catégorisent les IDS et proposent une taxonomie complète des approches basées sur des réseaux neuronaux superficiels et profonds. Cette étude analyse les performances des algorithmes de ML en détection d'anomalies, en soulignant le rôle crucial de la sélection des caractéristiques (feature selection) lors des phases de classification et d'entraînement. Elle examine également l'impact de cette sélection sur l'efficacité des IDS basés sur le ML, et conclut par une discussion sur les taux de faux positifs et de vrais positifs, offrant ainsi des pistes pour développer des systèmes plus fiables.

Les auteurs de [135] proposent une revue concise des IDS fondés sur le DL, en passant en revue divers algorithmes et aspects de la détection d'intrusion. Leur article inclut également une analyse critique de plusieurs jeux de données publics, mettant en lumière leurs forces et limites.

4.2.2 Applications des Réseaux Neuronaux Profonds

Farhan et al. [136] ont conçu un réseau neuronal dense entièrement connecté (Dense DNN) pour la détection d'intrusions basée sur les flux, en utilisant le jeu de données CSE-CIC-IDS2018 (disponible sur AWS). Leur modèle a atteint une précision de détection remarquable d'environ 90

Zhang et al. [137] ont introduit une technique innovante combinant des réseaux LSTM et des auto-encodeurs (AN) pour la détection d'intrusions. Après un prétraitement du jeu de données KDDcup99, ils ont utilisé un auto-encodeur pour réduire la dimensionnalité des données, puis extrait des caractéristiques pour entraîner un modèle LSTM. Les résultats ont montré une amélioration de 2% en précision et une réduction des taux de fausses alarmes par rapport aux méthodes traditionnelles.

Gamage et al. [138] ont comparé quatre modèles de DL (réseaux feed-forward, auto-encodeurs, Deep Belief Networks, et LSTM) sur des jeux de données historiques (KDD99, NSL-KDD) et contemporains (CIC-IDS2017, CIC-IDS2018). Les réseaux feed-forward supervisés ont obtenu les meilleures performances en termes de précision, F1-score et efficacité d'entraînement, surpassant les modèles semi-supervisés.

4.2.3 Détection Ciblée et Optimisation

Kanimozhi et al. [139] ont utilisé CSE-CIC-IDS2018 pour classer les attaques de botnets, une menace critique pour le secteur bancaire. Ils ont évalué plusieurs

classificateurs (Naive Bayes, k-NN, AdaBoost, Random Forest, SVM) et techniques d'IA, mettant en évidence leur efficacité différentielle.

Dans [140], les auteurs expliquent les concepts de base des réseaux de Boltzmann contraints (RBM) et des Deep Belief Networks (DBN) appliqués aux IDS, suivis d'une revue détaillée des modèles existants.

Abdallah et al. [141] ont développé une taxonomie des IDS intégrant des méthodes de ML supervisé. Leur analyse de quatre jeux de données (KDD'99, NSL-KDD, CICIDS2017, UNSW-NB15) confirme la robustesse des algorithmes supervisés pour la classification des intrusions.

4.2.4 Réduction de Dimension et Modèles Hybrides

Sarhan et al. [142] ont étudié l'impact de la PCA, des auto-encodeurs et de la LDA sur la précision de classification, en testant trois modèles de DL (DFF, CNN, RNN) et trois algorithmes classiques (régression logique, arbres de décision, Naive Bayes) sur UNSW-NB15, ToN-IoT et CSE-CIC-IDS2018. Leurs résultats identifient les combinaisons optimales de méthodes et le nombre idéal de dimensions pour chaque jeu de données.

Les auteurs de [143] ont proposé un modèle LSTM entraîné sur CSE-CIC-IDS2018, atteignant une précision de 99% pour la détection en temps réel. Hnamte et al. [144] ont conçu un IDS basé sur le DL, obtenant 100% de précision sur CICIDS2018 et 99.64% sur Edge-IIoT en classification multi-classes.

4.2.5 Approches Spécialisées et Hautes Performances

Elsayed et al. [145] ont développé SATIDS, un système à deux niveaux utilisant un réseau LSTM amélioré. Évalué sur ToN-IoT et InSDN, il a atteint 96.35% de précision sur le premier et 99.73% sur le second, avec des taux de détection et de précision supérieurs à 98%.

Saran et al. [146] ont conçu un IDS pour l'IoT en testant plusieurs classificateurs (k-NN, SVM, Random Forest) sur MQTT-IoT-IDS2020, avec des précisions allant jusqu'à 99.98% (Random Forest).

Enfin, Alzughaibi et al. [147] ont optimisé les IDS pour le cloud en combinant MLP avec backpropagation et optimisation par essaim particulaire (PSO). Leur modèle a atteint 98.41% en classification multi-classes et 98.97% en classification binaire sur CSE-CIC-IDS2018.

4.3 Méthodologie

Cette section décrit l'architecture du réseau utilisée dans notre expérience de classification multiclasse des attaques IDS.

4.3.1 Dataset

Le jeu de données CSE-CIC-IDS2018, disponible sur la plateforme AWS, constitue une ressource majeure pour la recherche en cybersécurité, offrant une collection complète de plus de 16 millions d'instances réseau caractérisées par 84 attributs détaillés et recueillies sur une période de 10 jours (du 14 février au 2 mars 2018). Ce dataset se distingue par sa reproduction fidèle d'environnements réseau réels, combinant à la fois du trafic normal et 13 catégories d'attaques sophistiquées (incluant DDoS, brute force, infiltration et injection SQL), le tout organisé en 10 fichiers distincts pour une gestion optimale. La richesse des métadonnées - comprenant des horodatages précis, des détails complets sur les paquets et les flux, ainsi que des motifs comportementaux caractéristiques - en fait un outil particulièrement précieux pour le développement et l'évaluation rigoureuse d'algorithmes avancés de détection d'intrusion. Son étiquetage minutieux (14 classes au total) et son échelle substantielle permettent non seulement d'entraîner des modèles complexes, mais aussi de mener des analyses comparatives fiables de solutions de sécurité. Accessible via AWS, ce dataset s'est imposé comme une référence incontournable dans la communauté scientifique, fournissant un cadre d'expérimentation réaliste pour tester des approches innovantes contre des menaces contemporaines, tout en facilitant l'intégration dans les workflows de recherche grâce à sa structure bien organisée et sa compatibilité avec les outils cloud modernes.

4.3.2 Prétraitement des données

Le prétraitement des données consiste à nettoyer, transformer et organiser les données brutes pour les adapter aux algorithmes de DL. Il comprend des techniques telles que le nettoyage des données, la gestion des valeurs manquantes et la normalisation, qui visent à garantir la cohérence, la précision et l'absence d'erreurs des données. Un prétraitement efficace améliore les performances des modèles de DL, ce qui permet d'obtenir des prédictions plus précises. Étant donné que l'ensemble de données CSE-CIC-IDS2018 est divisé en 10 fichiers, le traitement de chacun d'entre eux séparément serait fastidieux. Par conséquent, l'approche a consisté à fusionner les 10 fichiers en un seul afin de simplifier le traitement des données.

4.3.2.1 Fusion des fichiers

Avant d'entreprendre les opérations de prétraitement, une étape préliminaire cruciale a consisté à fusionner l'ensemble des dix fichiers constitutifs du dataset CSE-CIC-IDS2018 en un fichier unique, cette consolidation visant à optimiser l'efficacité du traitement ultérieur des données. Ce processus d'agrégation a cependant révélé deux problématiques techniques significatives nécessitant une intervention particulière. Premièrement, nous avons identifié d'importantes incohérences dans les types de données attribués aux différentes caractéristiques (features) selon les fichiers sources

- certaines variables apparaissant alternativement sous forme d'entiers ou de flottants dans certains fichiers tout en étant enregistrées comme objets (strings) dans d'autres, ce qui aurait gravement compromis la cohérence des analyses ultérieures. Pour résoudre ce problème, nous avons implémenté une standardisation systématique des types de données, garantissant ainsi une homogénéité parfaite sur l'ensemble du dataset fusionné. Deuxièmement, nous avons été confrontés à une limitation technique majeure avec l'impossibilité d'intégrer un fichier particulièrement volumineux (4.05 Go) en raison de contraintes mémoire. Face à cette contrainte, nous avons opté pour une stratégie de sélection raisonnée consistant à éliminer prioritairement une partie des instances bénignes (trafic normal) tout en conservant intégralement toutes les instances d'attaques - ces dernières présentant un intérêt analytique primordial pour notre étude. Il est important de noter que cette sélection a été opérée de manière à préserver un échantillon représentatif de trafic normal suffisant pour permettre des comparaisons statistiquement valides, tout en respectant les limites techniques de notre infrastructure. Bien que cette opération ait entraîné une réduction contrôlée du volume total de données, le dataset résultant conserve une richesse informationnelle et une représentativité tout à fait adéquates pour nos objectifs de recherche, avec une distribution préservée des différentes classes d'attaques et un ratio attaques/-trafic normal restant pertinent pour l'entraînement et l'évaluation de nos modèles. Cette phase préalable de consolidation et de normalisation s'est ainsi avérée essentielle pour garantir la qualité et l'exploitabilité optimale de notre base de données avant d'engager les étapes ultérieures de prétraitement plus approfondi.

4.3.2.2 Nettoyage des données

Le jeu de données a été nettoyé en éliminant les valeurs manquantes (NaN et INF) et les lignes en double pour éviter d'affecter négativement les performances du modèle. Les caractéristiques non pertinentes pour la détection d'attaques (comme Timestamp, Protocol, Flow ID) et celles contenant des valeurs nulles (telles que Bwd PSH Flags, Fwd Byts/b Avg) ont été exclues afin de prévenir les erreurs et d'améliorer la précision. Ce processus de nettoyage et de traitement a abouti à un ensemble de données affiné, prêt pour l'analyse. Il a notamment impliqué la gestion des valeurs manquantes ou incohérentes, la correction des erreurs et le traitement des valeurs aberrantes, améliorant ainsi la précision et la fiabilité du dataset. La visualisation des données nettoyées, comme illustré dans la Figure 4.1, permet de mieux comprendre les corrélations entre variables et les motifs essentiels pour la construction du modèle.

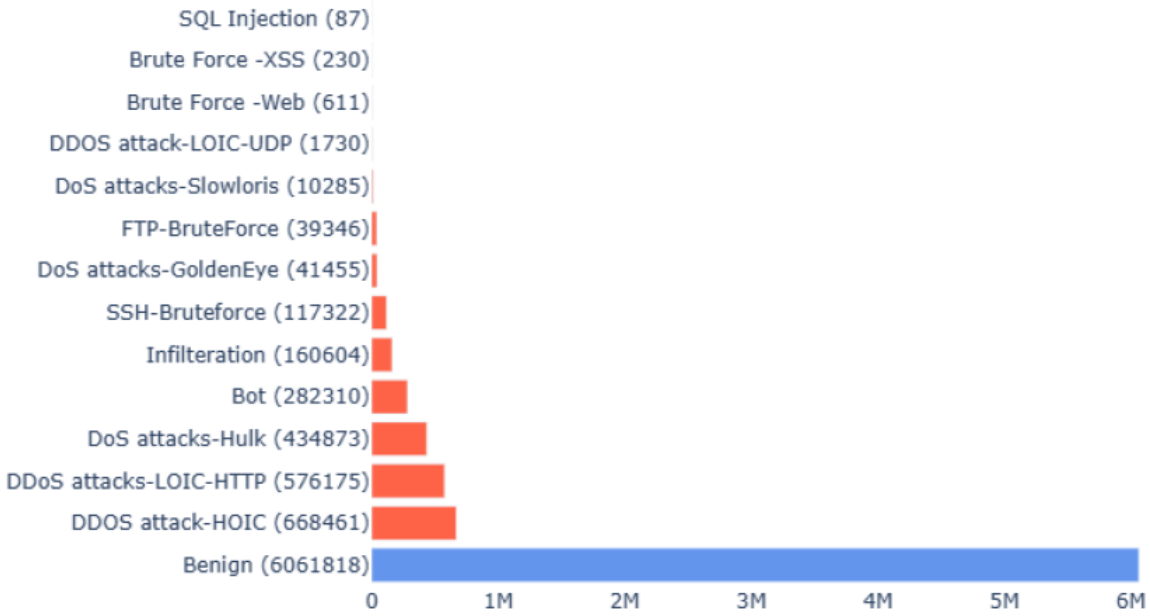
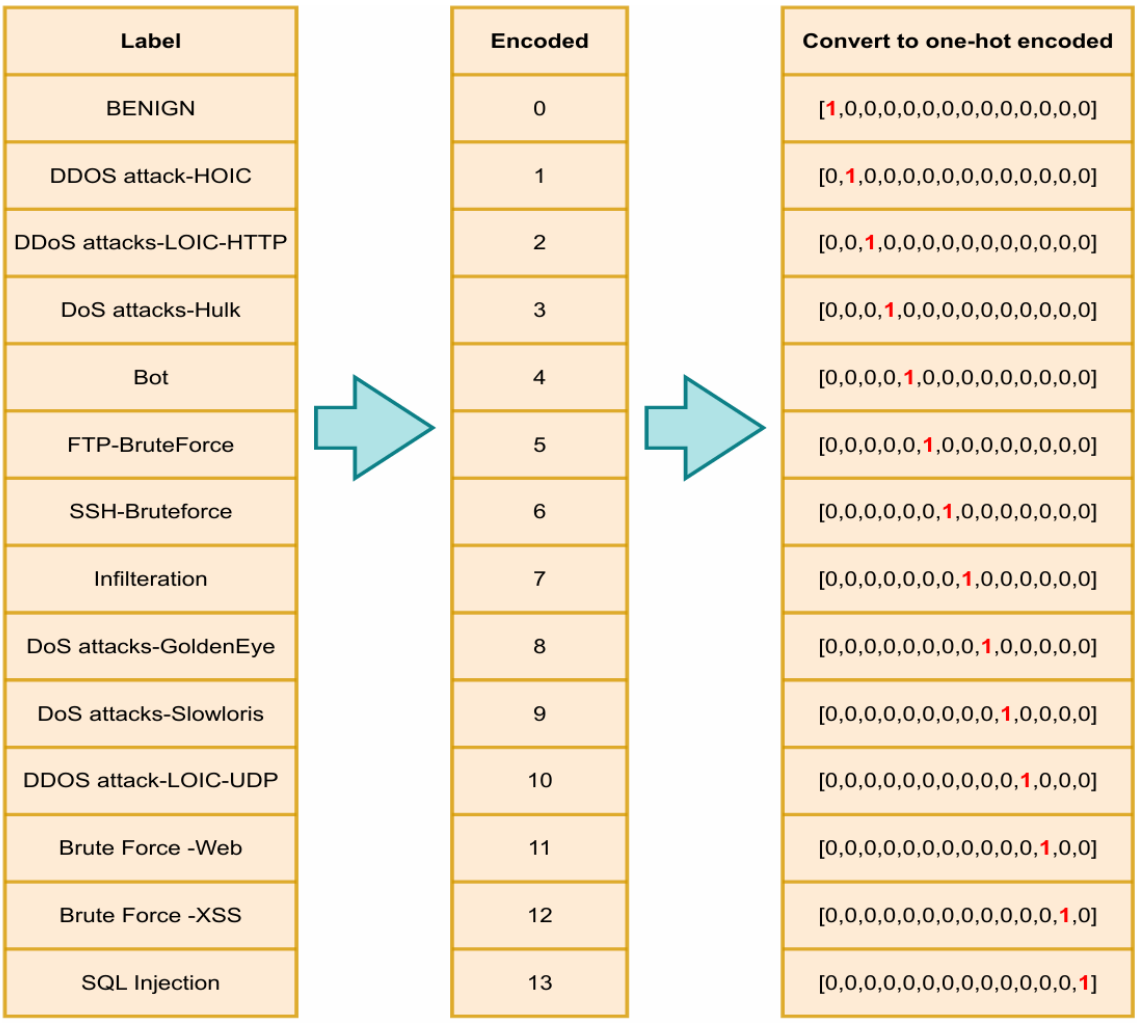


Figure 4.1 *– Distribution des étiquettes dans le dataset nettoyé.*

4.3.2.3 Codage des étiquettes (Label encoding)

Pour entraîner des modèles, les données doivent être représentées numériquement car les algorithmes de Machine Learning (ML) et Deep Learning (DL) nécessitent des matrices numériques. Les variables catégorielles, telles que les étiquettes des types de trafic réseau (par exemple, "benign" et "DDOS-attack-HOIC"), doivent être converties en format numérique. Cette conversion peut être réalisée via un encodage par étiquettes (label encoding), où chaque catégorie se voit attribuer une valeur numérique (par exemple, 0 pour "benign" et 1 pour "DDOS-attack-HOIC"), ou via un encodage one-hot [148], qui représente chaque étiquette sous forme de vecteur binaire (par exemple, "benign" comme [1, 0, 0, ...] et "DDOS-attack-HOIC" comme [0, 1, 0, ...]) - voir Figure 4.2. L'encodage one-hot évite de créer des hiérarchies artificielles entre les étiquettes. Après encodage, la colonne contenant les étiquettes encodées est isolée en tant que variable cible, distincte du reste du jeu de données.



Label	Encoded	Convert to one-hot encoded
BENIGN	0	[1,0,0,0,0,0,0,0,0,0,0,0,0,0]
DDOS attack-HOIC	1	[0,1,0,0,0,0,0,0,0,0,0,0,0,0]
DDoS attacks-LOIC-HTTP	2	[0,0,1,0,0,0,0,0,0,0,0,0,0,0]
DoS attacks-Hulk	3	[0,0,0,1,0,0,0,0,0,0,0,0,0,0]
Bot	4	[0,0,0,0,1,0,0,0,0,0,0,0,0,0]
FTP-BruteForce	5	[0,0,0,0,0,1,0,0,0,0,0,0,0,0]
SSH-Bruteforce	6	[0,0,0,0,0,0,1,0,0,0,0,0,0,0]
Infilteration	7	[0,0,0,0,0,0,0,1,0,0,0,0,0,0]
DoS attacks-GoldenEye	8	[0,0,0,0,0,0,0,0,1,0,0,0,0,0]
DoS attacks-Slowloris	9	[0,0,0,0,0,0,0,0,0,1,0,0,0,0]
DDOS attack-LOIC-UDP	10	[0,0,0,0,0,0,0,0,0,0,1,0,0,0]
Brute Force -Web	11	[0,0,0,0,0,0,0,0,0,0,0,1,0,0]
Brute Force -XSS	12	[0,0,0,0,0,0,0,0,0,0,0,0,1,0]
SQL Injection	13	[0,0,0,0,0,0,0,0,0,0,0,0,0,1]

Figure 4.2 *– Label encoding/one hot encoding.*

4.3.2.4 Normalisation

La normalisation des données constitue une étape cruciale de prétraitement pour l'analyse statistique et le machine learning. Elle permet de standardiser les données sur une échelle commune, facilitant ainsi les comparaisons et analyses entre variables. Cette opération révèle des motifs, valeurs aberrantes et relations qui pourraient rester cachés autrement, tout en améliorant les performances et l'interprétabilité des modèles en atténuant l'impact des variables à grande échelle et en réduisant les biais. Nous avons employé la normalisation min-max, qui redimensionne les données dans un intervalle spécifique (généralement entre 0 et 1) en soustrayant la valeur minimale et en divisant par l'étendue des données [149]. Cette méthode préserve la distribution originale des données tout en les ajustant à la plage souhaitée. Globalement, la normalisation s'avère essentielle dans notre processus de prétraitement, garantissant des résultats plus fiables et exploitables.

4.3.2.5 Division des données

En apprentissage profond (Deep Learning), la division des données en ensembles d'entraînement, de validation et de test est essentielle pour le développement des modèles. Cette approche permet d'évaluer les performances du modèle sur de nouvelles données invisibles et prévient le surapprentissage. Typiquement, le jeu de données est séparé en trois sous-ensembles : 90% pour l'entraînement, 10% pour le test, et 10% des données d'entraînement sont réservées pour la validation (voir Figure 4.3). L'ensemble de validation sert à évaluer les performances et à ajuster les hyperparamètres pendant l'entraînement. Le modèle est entraîné sur l'ensemble d'entraînement et évalué sur l'ensemble de validation. L'ensemble de test, conservé séparément jusqu'à la fin de l'entraînement, permet d'évaluer les performances du modèle sur des données totalement nouvelles. Cette méthode garantit une évaluation efficace du modèle et réduit les risques de surapprentissage.

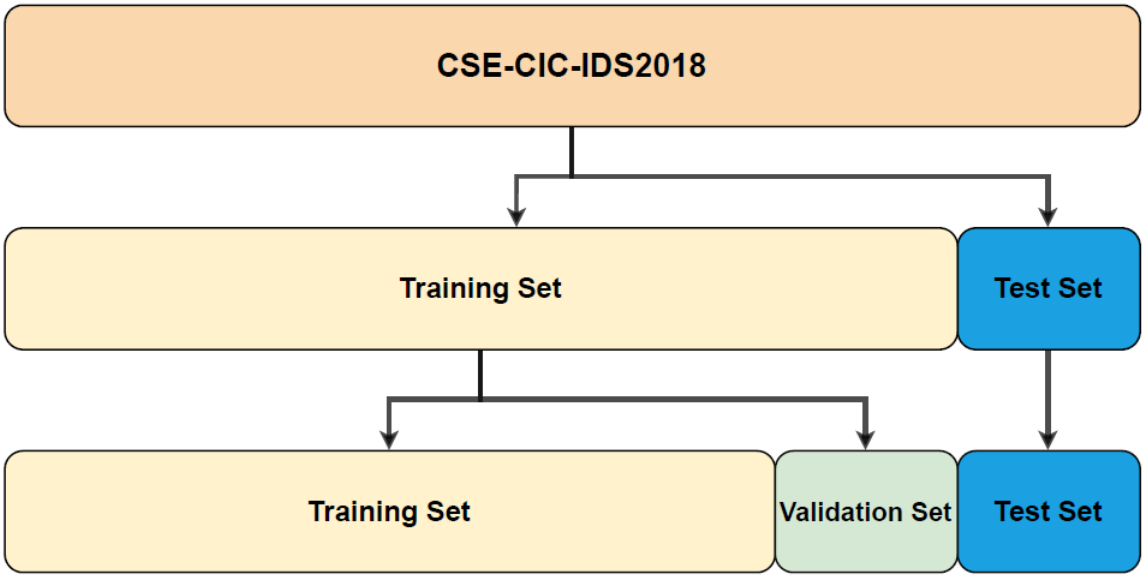


Figure 4.3 *– Division du dataset CIC-IDS2018 en trois parties.*

4.3.3 Métriques d'évaluation

Les paramètres suivants sont utilisés pour valider les modèles créés [150].

1. **Accuracy :** L'accuracy est une mesure de l'exactitude des prédictions positives d'un modèle. Elle est définie comme le rapport entre le nombre de prédictions correctes et le nombre total de prédictions effectuées par le modèle.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{4.1}$$

2. **Precision :** La proportion de prédictions correctes parmi les prédictions positives est appelée précision, ou valeur prédictive positive (en anglais, positive

predictive value (PPV)) :

$$Precision = \frac{TP}{TP + FP} \tag{4.2}$$

3. **Recall :** Nous appelons rappel (ou sensibilité) le taux de vrais positifs, c'est-à-dire la proportion d'exemples positifs correctement identifiés comme tels.

$$Recall = \frac{TP}{TP + FN} \tag{4.3}$$

4. **F-score :** Nous appelons score F (ou score F1) la moyenne harmonique de la précision et du rappel.

$$F - score = 2\frac{Precision \times Recall}{Precision + Recall} = \frac{2TP}{2TP + FP + FN} \tag{4.4}$$

5. **Specificity :** Nous appelons spécificité le taux de vrais négatifs, c'est-à-dire la proportion d'exemples négatifs correctement identifiés comme tels :

$$Specificity = \frac{TN}{FP + TN} \tag{4.5}$$

4.3.4 Création du modèle

Avant de développer le modèle, nous avons effectué plusieurs étapes préparatoires pour DL. Nous avons consolidé plusieurs fichiers en un seul ensemble de données, nettoyé les données en supprimant les doublons, les points non pertinents et en corrigeant les erreurs. Les variables catégorielles ont été converties en valeurs numériques à l'aide d'un encodage à un point. Nous avons normalisé les données pour nous assurer que toutes les caractéristiques étaient sur une échelle similaire, améliorant ainsi les performances de l'algorithme DL. Enfin, nous avons divisé les données en ensembles de formation et de test afin d'évaluer les performances du modèle sur de nouvelles données inédites.

Pour notre modèle DL, nous avons choisi le modèle séquentiel de la bibliothèque Keras [151]. Ce cadre est populaire pour sa flexibilité et sa conception conviviale. Le modèle séquentiel organise les couches dans une séquence linéaire, où chaque couche est ajoutée consécutivement. Il est idéal pour construire des modèles avec une seule entrée et une seule sortie, ce qui le rend simple et efficace pour nos besoins.

La conception et l'entraînement d'un NN impliquent plusieurs étapes critiques. La première étape consiste à sélectionner l'architecture du réseau national la mieux adaptée au problème. Dans notre cas, nous avons opté pour un réseau neuronal de type feedforward (FFNN). Une fois l'architecture choisie, l'étape suivante consiste à définir les hyperparamètres. Notre modèle comprend six couches : une couche d'entrée, quatre couches cachées et une couche de sortie. Plus précisément, la première couche cachée contient 48 neurones, la deuxième 32 neurones et les troisième et quatrième couches cachées 16 neurones chacune.

La couche de sortie se compose de 14 neurones, ce qui correspond au nombre de classes de sortie. Pour les couches cachées, nous utilisons la fonction d'activation ReLU (Rectified Linear Unit), qui est efficace pour modéliser les relations non linéaires dans les NN à progression directe. La couche de sortie utilise la fonction d'activation Softmax pour générer une distribution de probabilité entre les différentes classes, ce qui la rend adaptée aux tâches de classification multi-classes. Nous avons choisi l'optimiseur Adam, connu pour son efficacité dans le traitement de grands ensembles de données et sa capacité à converger vers des solutions optimales. La fonction de perte utilisée est la Crossentropie catégorielle, idéale pour mesurer l'écart entre les probabilités prédites et les étiquettes de classe réelles dans les scénarios de classification multi-classes. Pour atténuer le surajustement, nous avons appliqué une régularisation de poids L2 avec une valeur lambda de 0,0001, ce qui améliore les performances du modèle sur les données non vues.

Enfin, le modèle est entraîné à l'aide des données d'entraînement, au cours de laquelle les poids des neurones sont ajustés pour minimiser la fonction de perte. Ce processus est itératif et se poursuit jusqu'à ce que le modèle converge vers un ensemble de poids qui optimisent les performances sur les données d'apprentissage. La figure 4 illustre l'IDS proposé.

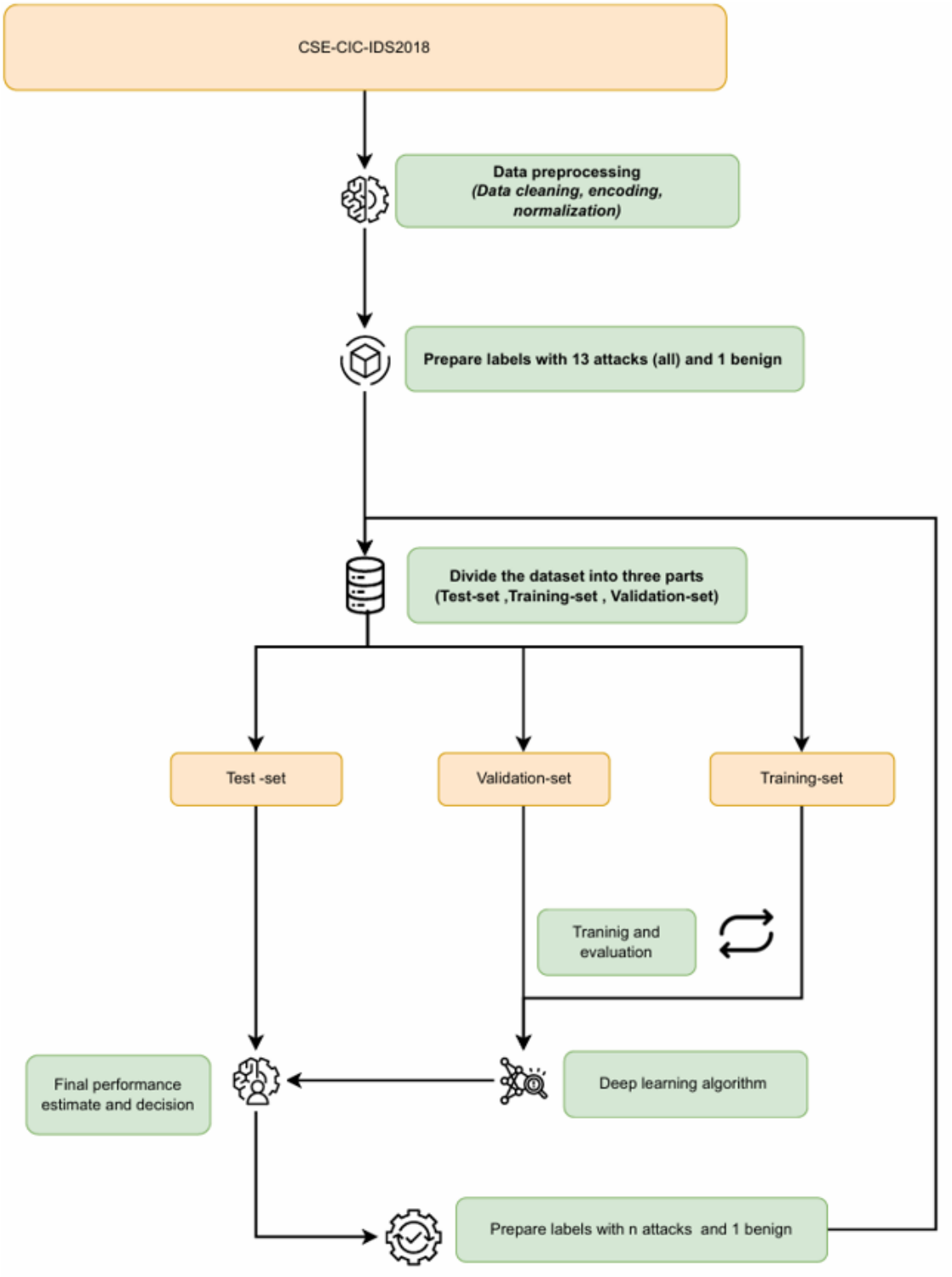


Figure 4.4 – *Architecture des composants du modèle proposé.*

4.4 Résultats et analyses

Après avoir entraîné le modèle de réseau de neurones (NN) avec une taille de lot de 2048 sur 50 itérations (epochs), il convient de mettre en évidence le déroulement du

processus d'apprentissage. Étant donné la taille importante du jeu de données, celui-ci a été divisé en lots, chacun contenant 2048 exemples d'apprentissage. La taille de lot constitue un hyperparamètre essentiel dans l'entraînement des réseaux de neurones, déterminant le nombre d'exemples utilisés à chaque itération d'apprentissage. Dans ce contexte, une séparation des données en validation a été réalisée avec un taux de 0,1, et un mélange aléatoire (shuffling) des données a été activé. Le mélange du jeu de données au début de chaque époque permet de randomiser l'ordre des exemples, évitant ainsi que le modèle ne s'adapte excessivement à une séquence particulière des données et favorisant l'apprentissage de représentations plus robustes.

Les Figures 4.5 et 4.6 présentent respectivement l'évolution de la précision et de la fonction de perte du modèle en fonction des époques au cours de l'entraînement.

Les résultats indiquent que le modèle a atteint un niveau de précision remarquable. Dès la première époque d'apprentissage, le modèle a atteint une précision de 95 %, démontrant sa capacité élevée à classer correctement les données. De plus, la stabilisation observée dès la quatrième époque suggère que le modèle a probablement convergé vers un ensemble fiable de paramètres, sans manifestation de surapprentissage (overfitting) ni de sous-apprentissage (underfitting).

Par ailleurs, une diminution rapide de la fonction de perte a été observée lors des premières itérations, suivie d'une stabilisation progressive au fur et à mesure de l'avancement de l'apprentissage. Cette évolution indique que le modèle a appris à effectuer des prédictions de plus en plus précises. La stabilisation constatée après quelques époques laisse penser que le modèle a atteint un niveau d'entraînement satisfaisant et qu'une amélioration supplémentaire significative est peu probable. Ces résultats témoignent de la capacité du modèle à apprendre efficacement et à s'ajuster de manière optimale aux données d'apprentissage.

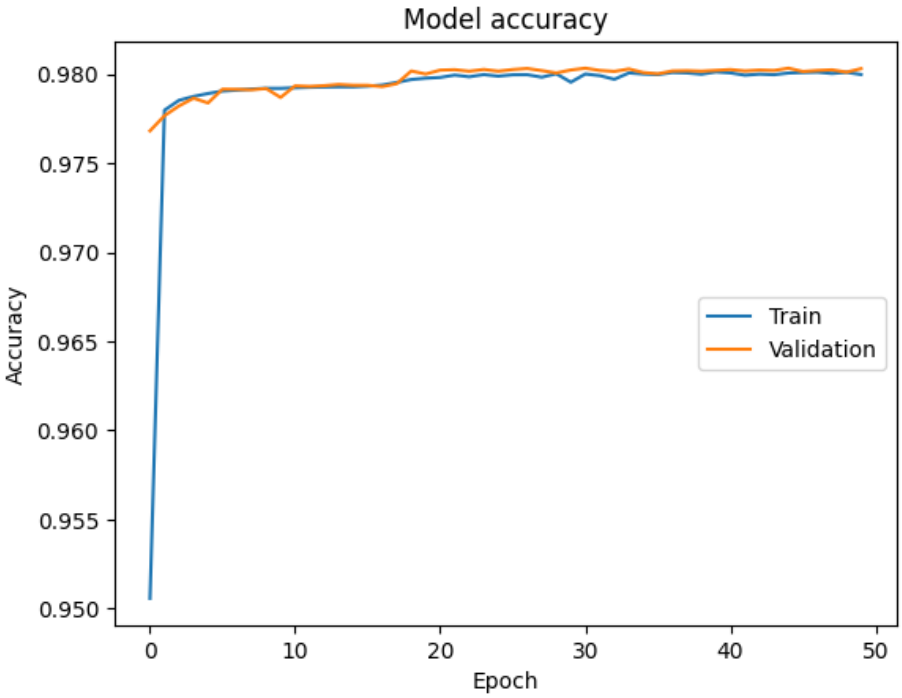


Figure 4.5 – *Training and Validation Accuracy.*

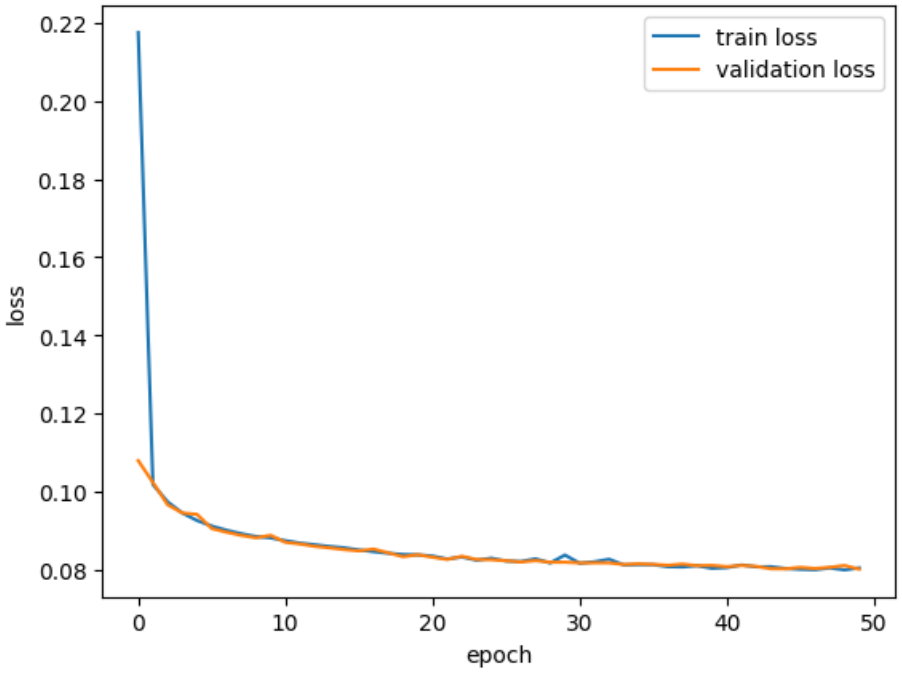


Figure 4.6 – *Training and Validation Loss.*

L'accuracy, la précision, le rappel (recall) et le score F1 se sont révélés être des métriques efficaces pour évaluer la performance de notre modèle. Ces indicateurs ont permis une évaluation globale et approfondie, nous permettant de prendre des

décisions éclairées quant au développement du modèle. Ils ont notamment facilité l'identification de pistes d'amélioration, telles que l'optimisation du seuil de décision pour la classification ou encore l'équilibrage de la distribution des classes au sein du jeu de données. Le Tableau 4.1 présente les valeurs de la précision, de la précision stricte, du rappel et du score F1 pour chacune des 14 classes, tandis que le Tableau 4.2 regroupe les valeurs globales de ces quatre métriques.

Tableau 4.1 – *Évaluation des métriques pour chaque classe*

Type d'attaque	**Accuracy**	**Precision**	**Recall**	**F1-score**
Benign	0.9993	0.97	1.00	0.99
DDOS attack-HOIC	1.00	1.00	1.00	1.00
DDoS attacks-LOIC-HTTP	0.9982	1.00	1.00	1.00
DoS attacks-Hulk	0.9997	1.00	1.00	1.00
Bot	0.9995	1.00	1.00	1.00
Infilteration	0.0083	0.44	0.01	0.02
SSH-Bruteforce	0.9997	1.00	1.00	1.00
DoS attacks-GoldenEye	0.9939	0.99	0.99	0.99
FTP-BruteForce	1.00	1.00	1.00	1.00
DoS attacks-Slowloris	0.9941	0.95	0.99	0.97
DDOS attack-LOIC-UDP	0.8361	0.80	0.84	0.82
Brute Force-Web	0.2777	0.91	0.28	0.43
Brute Force-XSS	0.5428	1.00	0.54	0.70
SQL Injection	0.00	0.00	0.00	0.00

Tableau 4.2 – *Métriques globales d'évaluation*

Accuracy	**Precision**	**Recall**	**F1-score**
98%	86.14%	76.07%	78.00%

Les valeurs plus faibles du rappel (recall) et du score F1, comparées à celles de la précision et de l'exactitude (accuracy), suggèrent que le modèle éprouve des difficultés à identifier correctement les vrais positifs et présente un taux plus élevé

de faux négatifs. Afin de mieux comprendre cette disparité, nous avons recours à la matrice de confusion, laquelle indique le nombre de vrais positifs, de vrais négatifs, de faux positifs et de faux négatifs. L'analyse de cette matrice de confusion (voir Figure 4.7) permet de repérer les points à améliorer et d'orienter les ajustements nécessaires dans la stratégie d'apprentissage, en vue d'optimiser la performance du modèle.

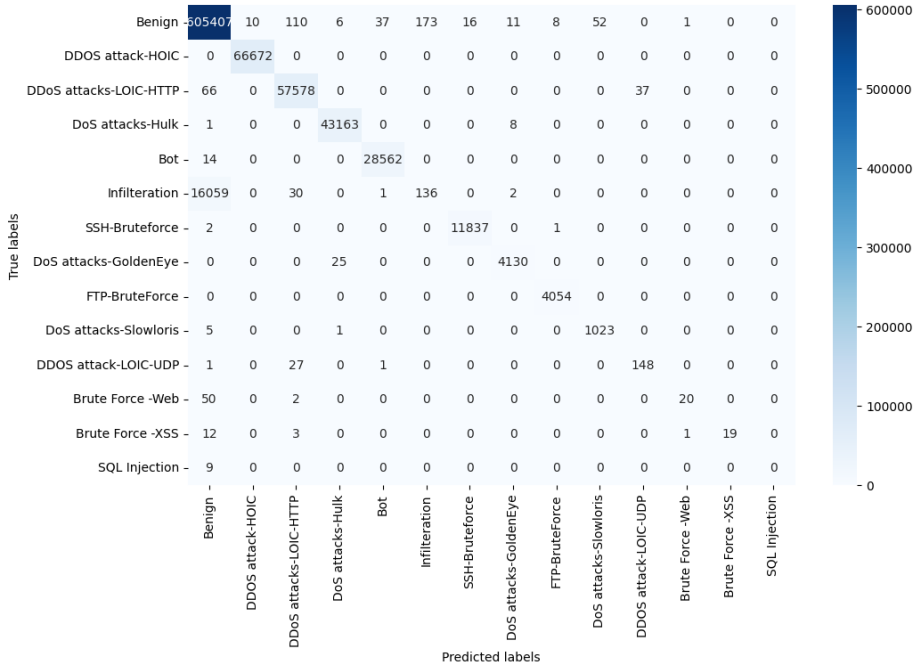


Figure 4.7 – *Matrice de confusion pour classification multi-classes.*

L'analyse de la matrice de confusion a révélé que les attaques de type SQL Injection, Brute Force-XSS et Brute Force-Web ont fréquemment été mal classées en tant que faux négatifs. De plus, bien que l'attaque DDOS attack-LOIC-UDP ait globalement été bien détectée, 30 % de ses échantillons ont été incorrectement classés comme faux positifs. Cette mauvaise classification résulte probablement d'un nombre insuffisant d'exemples d'apprentissage pour ces types d'attaques [152], limitant ainsi la capacité du modèle à apprendre et à identifier avec précision leurs schémas caractéristiques.

L'attaque Infiltration a quant à elle été classée à tort comme une absence d'attaque, malgré la présence d'un nombre suffisant d'exemples d'entraînement. Cette erreur s'explique vraisemblablement par la nature même de cette attaque [153], qui consiste à imiter le comportement humain normal afin de contourner les mécanismes de détection et d'obtenir un accès non autorisé.

Afin de remédier à ces difficultés, une approche alternative pourrait être envisagée : supprimer l'attaque Infiltration et regrouper les attaques DDOS attack-LOIC-UDP, Brute Force-Web, Brute Force-XSS et SQL Injection au sein d'une catégorie unique dénommée Autres Attaques (OA).

4.4.1 Approche améliorée

Sur la base des résultats précédemment obtenus, où nous avons pris en compte 13 types d'attaques ainsi que le trafic bénin, deux actions ont été entreprises. Premièrement, les quatre attaques suivantes : DDOS attack-LOIC-UDP, Brute Force-Web, Brute Force-XSS et SQL Injection ont été regroupées en une seule catégorie intitulée Autres Attaques (OA). Deuxièmement, l'attaque Infiltration a été entièrement supprimée. Cette décision a été motivée par les difficultés rencontrées pour obtenir des données supplémentaires. Le regroupement de ces attaques est supposé améliorer les performances du modèle.

Nous avons conservé notre approche initiale en y apportant une modification essentielle : la consolidation des quatre types d'attaques en une catégorie unique et l'exclusion de l'attaque Infiltration. Cette adaptation a conduit à des améliorations notables des différentes métriques, notamment une augmentation de l'exactitude (accuracy), de la précision (precision), du rappel (recall) et du score F1. De plus, le modèle a démontré de meilleures capacités de détection. Afin de fournir une analyse détaillée de nos résultats, nous avons inclus un examen approfondi de la matrice de confusion, de la précision, du score F1, du rappel et de l'exactitude pour chaque type d'attaque. Les Figures 4.8 et 4.9 présentent respectivement l'évolution de la précision et de la perte du modèle en fonction des époques au cours du processus d'entraînement.

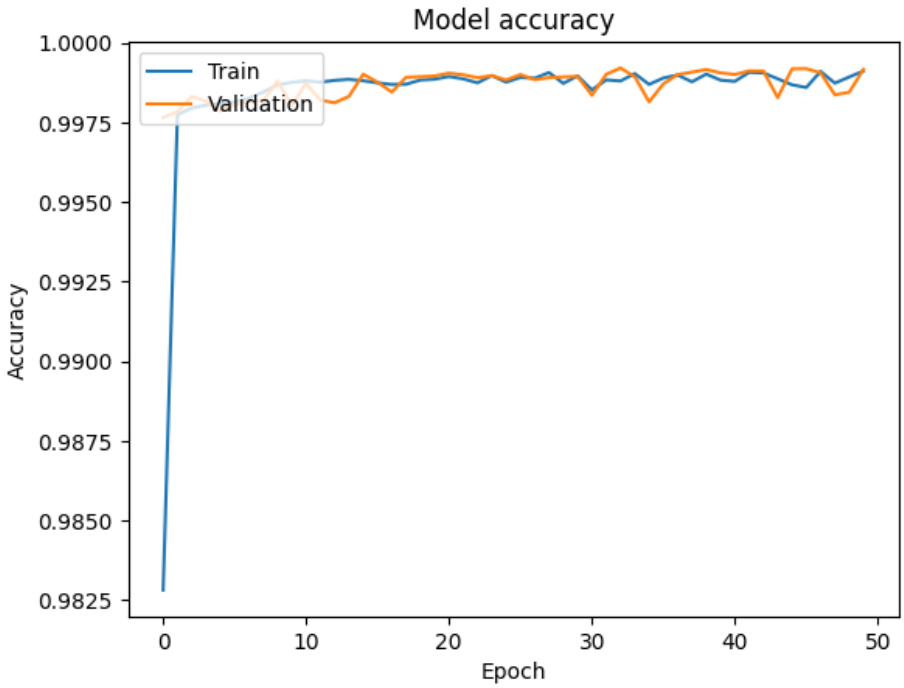


Figure 4.8 *– Training and Validation Accuracy - Improved approach.*

Le Tableau 4.3 présente les métriques d'évaluation pour chaque étiquette indivi-

duelle, tandis que le Tableau 4.4 fournit les métriques d'évaluation globales du modèle amélioré.

Tableau 4.3 *– Évaluation des métriques pour chaque étiquette*

Type d'attaque	**Accuracy**	**Precision**	**Recall**	**F1-score**
Benign	0.9996	0.9996	0.9996	0.9996
DDOS attack-HOIC	0.9998	1.00	0.9999	0.9999
DDoS attacks-LOIC-HTTP	0.9982	0.9962	0.9982	0.9972
DoS attacks-Hulk	0.9997	0.9960	0.9997	0.9979
Bot	0.9988	0.9991	0.9989	0.9990
SSH-Bruteforce	0.9997	0.9994	0.9997	0.9996
DoS attacks-GoldenEye	0.9595	0.9957	0.9596	0.9773
FTP-BruteForce	1.00	0.9973	1.00	0.9986
DoS attacks-Slowloris	0.9802	0.9586	0.9803	0.9693
OA	0.3825	0.9194	0.3826	0.5403

Tableau 4.4 *– Métriques globales d'évaluation*

Accuracy	**Precision**	**Recall**	**F1-score**
99.91%	98.61%	93.18%	94.78%

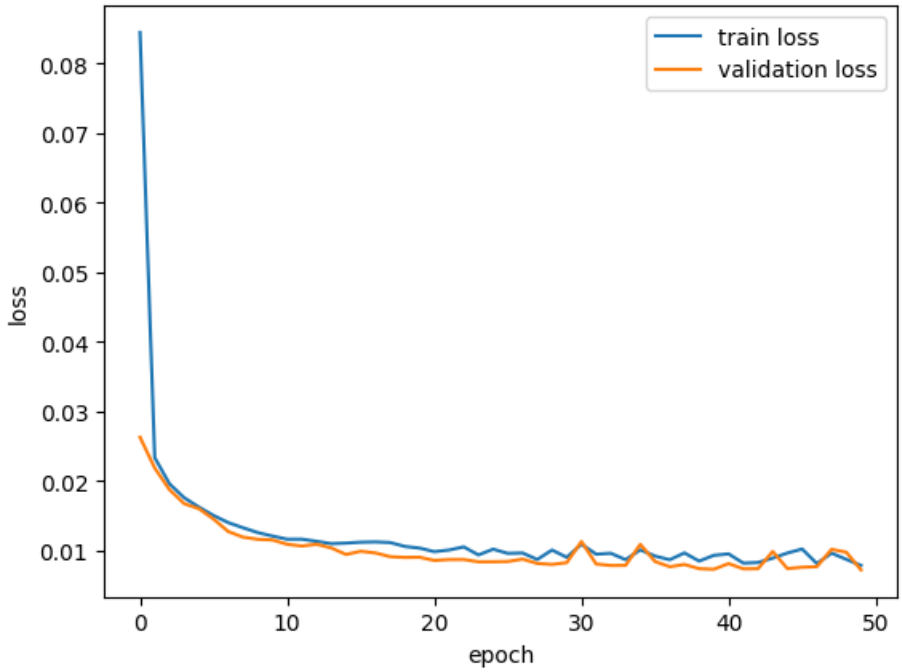


Figure 4.9 *– Training and Validation Loss - Improved approach.*

La Figure 4.10 présente la matrice de confusion dans le cas de la classification multiclasse pour le modèle amélioré.

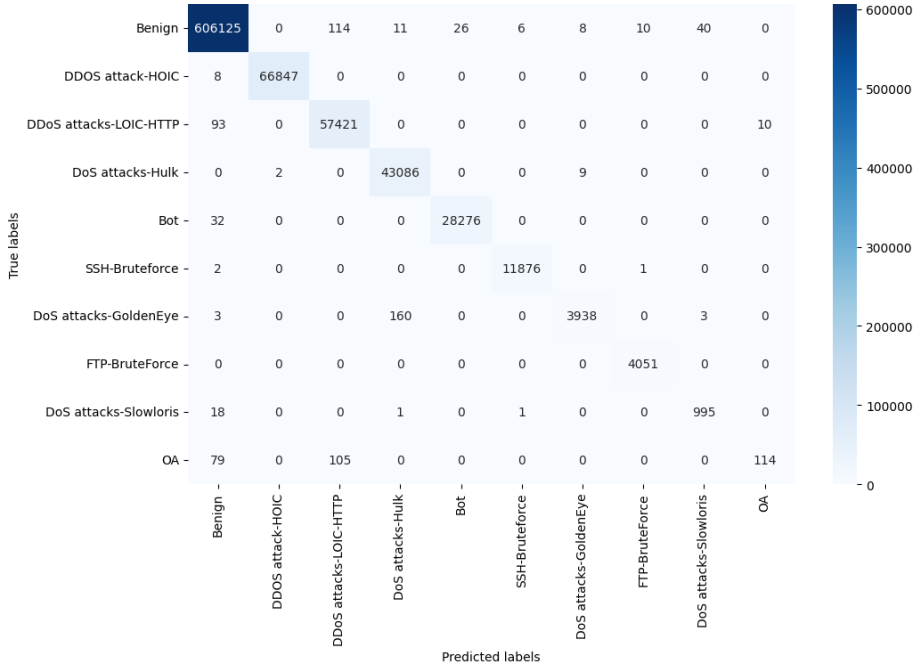


Figure 4.10 *– Matrice de confusion pour l'approche amélioré.*

4.5 Conclusion

La sécurité des réseaux est cruciale pour prévenir des problèmes graves tels que le vol de données, les interruptions de service ou les pertes financières. Pour renforcer cette sécurité, les administrateurs réseau se tournent de plus en plus vers des systèmes de détection d'intrusion (IDS) basés sur l'apprentissage profond (DL). Ce chapitre se concentre sur le développement d'un modèle DL pour détecter et classer les anomalies réseau à l'aide du jeu de données CSE-CIC-IDS2018. Les données ont été prétraitées via un nettoyage, une gestion des valeurs manquantes et une normalisation. Lors des tests initiaux, cinq attaques ont été mal détectées. Une approche révisée, consistant à supprimer l'attaque d'Infiltration mal identifiée et à regrouper quatre autres attaques dans une catégorie unique appelée "Autres Attaques", a donné des résultats améliorés. Bien que les performances globales soient satisfaisantes, l'étude a rencontré des limites dues à un volume de données insuffisant, soulignant la nécessité de jeux de données plus complets.

Conclusion générale

La sécurité des réseaux constitue aujourd'hui un enjeu fondamental, face à la multiplication des menaces informatiques capables d'engendrer des conséquences catastrophiques telles que le vol de données sensibles, des interruptions de service prolongées ou encore des pertes financières considérables. Les administrateurs de sécurité cherchent en permanence à mettre en uvre des solutions innovantes permettant d'assurer un environnement réseau hautement sécurisé. L'une des approches les plus prometteuses repose sur l'intégration de l'apprentissage profond dans les systèmes de détection d'intrusion (IDS).

Dans le cadre de ce travail de recherche, nous avons développé un modèle de détection et de classification des anomalies réseau en exploitant les techniques de Deep Learning. Le jeu de données CSE-CIC-IDS2018 a été sélectionné comme base d'entraînement, la qualité de cet ensemble de données jouant un rôle déterminant sur les performances des modèles entraînés. Afin de préparer les données à l'apprentissage, plusieurs étapes de prétraitement ont été appliquées, incluant le nettoyage des données, le traitement des valeurs manquantes et la normalisation. Les premières expérimentations ont révélé la difficulté du modèle à classer correctement certains types d'attaques lorsque toutes les classes étaient considérées simultanément. En conséquence, une approche alternative consistant à regrouper et supprimer certaines classes faiblement représentées a été adoptée, permettant d'améliorer significativement les performances du modèle. L'approche proposée s'est ainsi révélée satisfaisante en termes de fiabilité et d'efficacité.

L'une des principales limites rencontrées durant les expérimentations réside dans l'insuffisance de diversité des données d'entraînement, ce qui a parfois affecté la capacité de généralisation des modèles et la stabilité des performances obtenues. Ces observations soulignent l'importance de disposer de jeux de données plus variés et représentatifs pour améliorer la robustesse des systèmes de détection d'intrusion.

Perspectives de recherche

Afin d'aller plus loin dans l'amélioration des IDS basés sur le Deep Learning, plusieurs perspectives peuvent être envisagées. D'abord, l'exploration de techniques avancées telles que l'apprentissage par adversaires (Adversarial Training) ou les mo-

dèles d'ensemble (Ensemble Learning) pourrait renforcer la résilience des modèles face aux attaques sophistiquées. Ensuite, l'intégration des IDS avec d'autres mécanismes de protection (pare-feux intelligents, honeypots, systèmes de prévention d'intrusion IDPS) permettrait de bâtir des architectures de sécurité multicouches plus robustes.

Par ailleurs, le développement de modèles hybrides combinant des approches de Machine Learning supervisé et non supervisé avec des techniques basées sur des règles explicites pourrait améliorer la détection d'anomalies rares ou inconnues. L'intégration de l'intelligence artificielle explicable (Explainable AI, XAI) constituerait également un axe prometteur, permettant d'améliorer l'interprétabilité et la transparence des décisions prises par les modèles de Deep Learning, condition essentielle pour leur adoption dans des contextes critiques.

Enfin, des travaux futurs pourraient s'orienter vers la conception de systèmes IDS distribués et fédérés, exploitant l'apprentissage fédéré pour préserver la confidentialité des données tout en assurant un apprentissage collaboratif à large échelle dans des environnements distribués tels que les infrastructures cloud, les réseaux 5G, ou les systèmes IoT et IIoT.

En définitive, en perfectionnant les IDS intelligents à l'aide des techniques d'apprentissage profond et des approches hybrides, il sera possible de proposer des solutions toujours plus efficaces et adaptatives pour contrer l'évolution rapide des menaces cybernétiques.

Liste des publications/communications

- International peer-reviewed journals

[1] Al Baraa Bouidaine, Djilali Moussaoui, Wafaa Ferhi, Mourad Hadjila, and Mohammed Hicham Hachemi, " Deep Learning-Based Anomaly and Intrusion Detection Using the CSE-CIC-IDS2018 Dataset", Engineering, Technology & Applied Science Research, Vol. 15, No. 4, p. 24782-24787, August 2025, https ://-doi.org/10.48084/etasr.11173

[2] M. Hadjila, M. Merzoug, W. Ferhi, D. Moussaoui, A.B. Bouidaine, M.H. Hachemi, "Obfuscated malware detection using deep neural network with ANOVA feature selection on CIC-MalMem-2022 dataset", Scientific and Technical Journal of Information Technologies, Mechanics and Optics, 2024, vol. 24, no 5, p. 849-857 doi : 10.17586/2226-1494-2024-24-5-849-857

[3] Wafaa Ferhi, Djilali Moussaoui, Mourad Hadjila, Al Baraa Bouidaine, Anomaly detection for IIoT : analyzing Edge-IIoTset dataset with varied class distributions, Scientific and Technical Journal of Information Technologies, Mechanics and Optics, Vol. 25, No.5, p. 876-887, September-October 2025. doi : 10.17586/2226-1494-2025-25-5-876-887

- International conferences with peer review

[4] Al Baraa Bouidaine, Wafaa Ferhi, Djilali Moussaoui, Mourad Hadjila, 'Heart Disease Evaluation using Deep Learning Techniques', International Congress on Health Sciences and Medical Technologies (ICHSMT 23), (Online) Tlemcen Algeria, 26-28 December 2023.

[5] Bouidaine Al Baraa, FERHI Wafaa , HADJILA Mourad , MOUSSAOUI Djillali, "Deep Learning Classifier for DDoS Attacks Detection Across CSE-CIC-IDS2018 Dataset", in First International Conference on Artificial Intelligence and Sustainable Development (ICAISD25), held on April 12th-13th, 2025, Ahmed Zabana University-Relizane, Algeria

[6] FERHI, Wafaa, HADJILA, Mourad, DJILLALI, Djilali Moussaoui, and Al Baraa Bouidaine. Machine Learning-based Classification of Diabetes Disease : A Case

Study with Orange Data Mining. In : 2023 International Conference on Electrical Engineering and Advanced Technology (ICEEAT). IEEE, 2023. p. 1-6.

[7] W. Ferhi, M. Hadjila, D. Moussaoui and A. B. Bouidaine, "Anomaly Detection in IoT : State-of-the-Art Techniques and Implementation Insights," 2024 2nd International Conference on Electrical Engineering and Automatic Control (ICEEAC), Setif, Algeria, 2024, pp. 1-7, doi : 10.1109/ICEEAC61226.2024.10576293.

- National conferences

[8] FERHI, Wafaa, HADJILA, Mourad, DJILLALI, Djilali Moussaoui, and Al Baraa Bouidaine. "IoT Anomaly Detection Strategies : A Roadmap for Effective Research and Implementation", Algerian Doctoral Conference on Computer Science ADCCS2024.

Bibliographie

[1] A. Kuzior, I. Tiutiunyk, A. Zielińska, and R. Kelemen, "Cybersecurity and cybercrime : Current trends and threats." *Journal of International Studies (2071-8330)*, vol. 17, no. 2, 2024.

[2] W. Stallings, *Network security essentials : applications and standards*. Pearson Education India, 2003.

[3] K. Scarfone, P. Mell *et al.*, "Guide to intrusion detection and prevention systems (idps)," *NIST special publication*, vol. 800, no. 2007, p. 94, 2007.

[4] P. Garcia-Teodoro, J. Diaz-Verdejo, G. Maciá-Fernández, and E. Vázquez, "Anomaly-based network intrusion detection : Techniques, systems and challenges," *computers & security*, vol. 28, no. 1-2, pp. 18–28, 2009.

[5] A. L. Buczak and E. Guven, "A survey of data mining and machine learning methods for cyber security intrusion detection," *IEEE Communications surveys & tutorials*, vol. 18, no. 2, pp. 1153–1176, 2015.

[6] R. Zuech, T. M. Khoshgoftaar, and R. Wald, "Intrusion detection and big heterogeneous data : a survey," *Journal of Big Data*, vol. 2, no. 1, p. 3, 2015.

[7] J. F. Kurose and K. W. Ross, *Computer networking : A top-down approach*. Pearson Harlow, England Boston, 2019.

[8] O. M. C. Osazuwa, O. Mitchell, and C. Osazuwa, "Confidentiality, integrity, and availability in network systems : A review of related literature," *International Journal of Innovative Science and Research Technology*, vol. 8, no. 12, pp. 1946–1953, 2023.

[9] P. Kumar and S. B. Rana, "Development of modified aes algorithm for data security," *Optik*, vol. 127, no. 4, pp. 2341–2345, 2016.

[10] J. Gondaliya, S. Savani, V. S. Dhaduvai, and G. Hossain, "Hybrid security rsa algorithm in application of web service," in *2018 1st International Conference on Data Intelligence and Security (ICDIS)*. IEEE, 2018, pp. 149–152.

[11] A. Ometov, S. Bezzateev, N. Mäkitalo, S. Andreev, T. Mikkonen, and Y. Koucheryavy, "Multi-factor authentication : A survey," *Cryptography*, vol. 2, no. 1, p. 1, 2018.

[12] R. S. Sandhu, "Role-based access control," in *Advances in computers*. Elsevier, 1998, vol. 46, pp. 237–286.

[13] A. Amiruddin, H. G. Afiansyah, and H. A. Nugroho, "Cyber-risk management planning using nist csf v1. 1, nist sp 800-53 rev. 5, and cis controls v8," in *2021 International Conference on Informatics, Multimedia, Cyber and Information System (ICIMCIS)*. IEEE, 2021, pp. 19–24.

[14] G. Culot, G. Nassimbeni, M. Podrecca, and M. Sartor, "The iso/iec 27001 information security management standard : literature review and theory-based research agenda," *The TQM Journal*, vol. 33, no. 7, pp. 76–105, 2021.

[15] L. O. Gostin, L. A. Levit, and S. J. Nass, "Beyond the hipaa privacy rule : enhancing privacy, improving health through research," 2009.

[16] B. Nadji, "Data security, integrity, and protection," in *Data, Security, and Trust in Smart Cities*. Springer, 2024, pp. 59–83.

[17] R. Dastres and M. Soori, "A review in recent development of network threats and security measures," *International Journal of Information Sciences and Computer Engineering*, 2021.

[18] G. Carl, G. Kesidis, R. R. Brooks, and S. Rai, "Denial-of-service attack-detection techniques," *IEEE Internet computing*, vol. 10, no. 1, pp. 82–89, 2006.

[19] E. A. Asonye, I. Anwuna, and S. M. Musa, "Securing zigbee iot network against hulk distributed denial of service attack," in *2020 IEEE 17th International Conference on Smart Communities : Improving Quality of Life Using ICT, IoT and AI (HONET)*. IEEE, 2020, pp. 156–162.

[20] T. Shorey, D. Subbaiah, A. Goyal, A. Sakxena, and A. K. Mishra, "Performance comparison and analysis of slowloris, goldeneye and xerxes ddos attack tools," in *2018 International Conference on Advances in Computing, Communications and Informatics (ICACCI)*. IEEE, 2018, pp. 318–322.

[21] G. A. Jaafar, S. M. Abdullah, and S. Ismail, "Review of recent detection methods for http ddos attack," *Journal of Computer Networks and Communications*, vol. 2019, no. 1, p. 1283472, 2019.

[22] V. Kumar, K. Kumar *et al.*, "Classification of ddos attack tools and its handling techniques and strategy at application layer," in *2016 2nd International Conference on Advances in Computing, Communication, & Automation (ICACCA)(Fall)*. IEEE, 2016, pp. 1–6.

[23] K. Apostol, "Brute-force attack," 2012.

[24] J. Hancock, T. M. Khoshgoftaar, and J. L. Leevy, "Detecting ssh and ftp brute force attacks in big data," in *2021 20th IEEE international conference on machine learning and applications (ICMLA)*. IEEE, 2021, pp. 760–765.

[25] S. K. Wanjau, G. M. Wambugu, and G. N. Kamau, "Ssh-brute force attack detection model based on deep learning," 2021.

[26] R. Hofstede, M. Jonker, A. Sperotto, and A. Pras, "Flow-based web application brute-force attack and compromise detection," *Journal of network and systems management*, vol. 25, no. 4, pp. 735–758, 2017.

[27] M. A. Vairagade, D. Sable, and V. Wadhankar, "Detecting and preventing sql injection and xss attack using web security mechanisms," in *International Conference on Modern Trends in Engineering Science and Technology (ICMTEST 2016)*, vol. 2, no. 5, pp. 06–11.

[28] M. Alghawazi, D. Alghazzawi, and S. Alarifi, "Detection of sql injection attack using machine learning techniques : a systematic literature review," *Journal of Cybersecurity and Privacy*, vol. 2, no. 4, pp. 764–777, 2022.

[29] W. Suttapak, J. Zhang, and L. Zhang, "Diminishing-feature attack : The adversarial infiltration on visual tracking," *Neurocomputing*, vol. 509, pp. 21–33, 2022.

[30] I. Ali, A. I. A. Ahmed, A. Almogren, M. A. Raza, S. A. Shah, A. Khan, and A. Gani, "Systematic literature review on iot-based botnet attack," *IEEE access*, vol. 8, pp. 212 220–212 232, 2020.

[31] C. Elliott, "Botnets : To what extent are they a threat to information security ?" *Information security technical report*, vol. 15, no. 3, pp. 79–103, 2010.

[32] S. William, *Computer security : principles and practice*. Pearson Education India, 2008.

[33] A. X. Liu and M. G. Gouda, "Diverse firewall design," *IEEE Transactions on parallel and distributed systems*, vol. 19, no. 9, pp. 1237–1251, 2008.

[34] Q. Zhang, "An overview and analysis of hybrid encryption : the combination of symmetric encryption and asymmetric encryption," in *2021 2nd international conference on computing and data science (CDS)*. IEEE, 2021, pp. 616–622.

[35] S. H. Standard, "Secure hash standard," *FIPS PUB*, pp. 180–1, 1995.

[36] G. González-Granadillo, S. González-Zarzosa, and R. Diaz, "Security information and event management (siem) : analysis, trends, and usage in critical infrastructures," *Sensors*, vol. 21, no. 14, p. 4759, 2021.

[37] M. Rakhra, B. Kaur, G. Aggarwal, D. Ather, R. Kler, and K. Jairath, "The zero trust paradigm : Revolutionizing network security," in *2025 International Conference on Networks and Cryptology (NETCRYPT)*. IEEE, 2025, pp. 1714–1719.

[38] S. Subramani, A. Kavitha, and A. R. Safrin, "Zero trust network architecture for modern enterprise environments," in *2025 International Conference on Data Science, Agents & Artificial Intelligence (ICDSAAI)*. IEEE, 2025, pp. 1–6.

[39] Z. M. Azmi, I. Fadhil, D. D. P. Baron, and A. M. Shiddiqi, "Zero trust network access (ztna) to secure website applications based on iso 25023," in *2025*

International Conference on Advancement in Data Science, E-learning and Information System (ICADEIS). IEEE, 2025, pp. 1–6.

[40] R. Di Pietro and L. V. Mancini, *Intrusion detection systems*. Springer Science & Business Media, 2008, vol. 38.

[41] S. Kumar, S. Gupta, and S. Arora, "Research trends in network-based intrusion detection systems : A review," *Ieee Access*, vol. 9, pp. 157 761–157 779, 2021.

[42] S. Jose, D. Malathi, B. Reddy, and D. Jayaseeli, "A survey on anomaly based host intrusion detection system," in *Journal of Physics : Conference Series*, vol. 1000. IOP Publishing, 2018, p. 012049.

[43] E. M. Maseno, Z. Wang, and H. Xing, "A systematic review on hybrid intrusion detection system," *Security and Communication Networks*, vol. 2022, no. 1, p. 9663052, 2022.

[44] A. Gangwar and S. Sahu, "A survey on anomaly and signature based intrusion detection system (ids)," *International Journal of Engineering Research and Applications*, vol. 4, no. 4, pp. 67–72, 2014.

[45] A. A. Ramaki and R. E. Atani, "A survey of it early warning systems : architectures, challenges, and solutions," *Security and Communication Networks*, vol. 9, no. 17, pp. 4751–4776, 2016.

[46] I. Goodfellow, Y. Bengio, A. Courville, and Y. Bengio, *Deep learning*. MIT press Cambridge, 2016, vol. 1, no. 2.

[47] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *nature*, vol. 521, no. 7553, pp. 436–444, 2015.

[48] C. M. Bishop and N. M. Nasrabadi, *Pattern recognition and machine learning*. Springer, 2006, vol. 4, no. 4.

[49] A. Esteva, B. Kuprel, R. A. Novoa, J. Ko, S. M. Swetter, H. M. Blau, and S. Thrun, "Dermatologist-level classification of skin cancer with deep neural networks," *nature*, vol. 542, no. 7639, pp. 115–118, 2017.

[50] E. Topol, *Deep medicine : how artificial intelligence can make healthcare human again*. Hachette UK, 2019.

[51] J. B. Heaton, N. G. Polson, and J. H. Witte, "Deep learning for finance : deep portfolios," *Applied Stochastic Models in Business and Industry*, vol. 33, no. 1, pp. 3–12, 2017.

[52] S. Zhang, L. Yao, A. Sun, and Y. Tay, "Deep learning based recommender system : A survey and new perspectives," *ACM computing surveys (CSUR)*, vol. 52, no. 1, pp. 1–38, 2019.

[53] S. Grigorescu, B. Trasnea, T. Cocias, and G. Macesanu, "A survey of deep learning techniques for autonomous driving," *Journal of field robotics*, vol. 37, no. 3, pp. 362–386, 2020.

[54] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert : Pre-training of deep bidirectional transformers for language understanding," in *Proceedings of the 2019 conference of the North American chapter of the association for computational linguistics : human language technologies, volume 1 (long and short papers)*, 2019, pp. 4171–4186.

[55] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," *Advances in neural information processing systems*, vol. 25, 2012.

[56] L. P. Kaelbling, M. L. Littman, and A. W. Moore, "Reinforcement learning : A survey," *Journal of artificial intelligence research*, vol. 4, pp. 237–285, 1996.

[57] Z. Ghahramani, "Unsupervised learning," in *Summer school on machine learning*. Springer, 2003, pp. 72–112.

[58] P. Cunningham, M. Cord, and S. J. Delany, "Supervised learning," in *Machine learning techniques for multimedia : case studies on organization and retrieval*. Springer, 2008, pp. 21–49.

[59] M. F. A. Hady and F. Schwenker, "Semi-supervised learning," *Handbook on neural information processing*, pp. 215–239, 2013.

[60] T. Chakraborty and U. Kumar, "Loss function," in *Encyclopedia of Mathematical Geosciences*. Springer, 2022, pp. 1–6.

[61] A. Singh, N. Thakur, and A. Sharma, "A review of supervised machine learning algorithms," in *2016 3rd international conference on computing for sustainable global development (INDIACom)*. Ieee, 2016, pp. 1310–1315.

[62] S. Ray, "A quick review of machine learning algorithms," in *2019 International conference on machine learning, big data, cloud and parallel computing (COMITCon)*. IEEE, 2019, pp. 35–39.

[63] S. F. Ahmed, M. S. B. Alam, M. Hassan, M. R. Rozbu, T. Ishtiak, N. Rafa, M. Mofijur, A. Shawkat Ali, and A. H. Gandomi, "Deep learning modelling techniques : current progress, applications, advantages, and challenges," *Artificial Intelligence Review*, vol. 56, no. 11, pp. 13 521–13 617, 2023.

[64] A. Gulli and S. Pal, *Deep learning with Keras*. Packt Publishing Ltd, 2017.

[65] M. Moocarme, M. Abdolahnejad, and R. Bhagwat, *The Deep Learning with Keras Workshop : Learn how to define and train neural network models with just a few lines of code*. Packt Publishing Ltd, 2020.

[66] R. Indrakumari, T. Poongodi, and K. Singh, "Introduction to deep learning," in *Advanced deep learning for engineers and scientists : a practical approach*. Springer, 2021, pp. 1–22.

[67] V. K. Vishnoi, N. R. Chauhan, and K. Kumar, *An introduction to deep learning*. Xoffencerpublication, 2024.

[68] C. Eliasmith, *How to build a brain : A neural architecture for biological cognition*. OUP USA, 2013.

[69] A. Krogh, "What are artificial neural networks ?" *Nature biotechnology*, vol. 26, no. 2, pp. 195–197, 2008.

[70] S. Walczak, "Artificial neural networks," in *Advanced methodologies and technologies in artificial intelligence, computer simulation, and human-computer interaction*. IGI Global Scientific Publishing, 2019, pp. 40–53.

[71] Z. Zhang, "Artificial neural network," in *Multivariate time series analysis in climate and environmental research*. Springer, 2017, pp. 1–35.

[72] M. H. Sazlı, "A brief review of feed-forward neural networks," *Communications Faculty of Sciences University of Ankara Series A2-A3 Physical Sciences and Engineering*, vol. 50, no. 01, 2006.

[73] G. Bebis and M. Georgiopoulos, "Feed-forward neural networks," *Ieee Potentials*, vol. 13, no. 4, pp. 27–31, 1994.

[74] B. J. Wythoff, "Backpropagation neural networks : a tutorial," *Chemometrics and Intelligent Laboratory Systems*, vol. 18, no. 2, pp. 115–155, 1993.

[75] M. Buscema, "Back propagation neural networks," *Substance use & misuse*, vol. 33, no. 2, pp. 233–270, 1998.

[76] L. Ma and K. Khorasani, "Facial expression recognition using constructive feedforward neural networks," *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 34, no. 3, pp. 1588–1595, 2004.

[77] A. Tetickovic and S. Klancnik, "Voice recognition with feed-forward neural network," in *Proc. 14th International Electrotechnical and Computer Science Conference ERK*, vol. 5, 2005, pp. 387–388.

[78] Y. Zhou, "Natural language processing with improved deep learning neural networks," *Scientific programming*, vol. 2022, no. 1, p. 6028693, 2022.

[79] P. Kumar, G. R. Seshadri, A. Hariharan, V. Mohandas, and P. Balasubramanian, "Financial market prediction using feed forward neural network," in *Technology Systems and Management : First International Conference, ICTSM 2011, Mumbai, India, February 25-27, 2011. Selected Papers*. Springer, 2011, pp. 77–84.

[80] F. Amato, A. López, E. M. Peña-Méndez, P. Vaňhara, A. Hampl, and J. Havel, "Artificial neural networks in medical diagnosis," pp. 47–58, 2013.

[81] A.-N. Sharkawy, "Forward and inverse kinematics solution of a robotic manipulator using a multilayer feedforward neural network," *Journal of Mechanical and Energy Engineering*, vol. 6, no. 2, 2022.

[82] Z. Li, F. Liu, W. Yang, S. Peng, and J. Zhou, "A survey of convolutional neural networks : analysis, applications, and prospects," *IEEE transactions on neural networks and learning systems*, vol. 33, no. 12, pp. 6999–7019, 2021.

[83] A. Ajit, K. Acharya, and A. Samanta, "A review of convolutional neural networks," in *2020 international conference on emerging trends in information technology and engineering (ic-ETITE)*. IEEE, 2020, pp. 1–5.

[84] A. L. Caterini and D. E. Chang, "Recurrent neural networks," in *Deep neural networks in a mathematical framework*. Springer, 2018, pp. 59–79.

[85] F. M. Salem, *Recurrent neural networks*. Springer, 2022.

[86] S. Sharma, S. Sharma, and A. Athaiya, "Activation functions in neural networks," *Towards Data Sci*, vol. 6, no. 12, pp. 310–316, 2017.

[87] H. Pratiwi, A. P. Windarto, S. Susliansyah, R. R. Aria, S. Susilowati, L. K. Rahayu, Y. Fitriani, A. Merdekawati, and I. R. Rahadjeng, "Sigmoid activation function in selecting the best model of artificial neural networks," in *Journal of physics : conference series*, vol. 1471, no. 1. IOP Publishing, 2020, p. 012010.

[88] Y. Bai, "Relu-function and derived function review," in *SHS web of conferences*, vol. 144. EDP Sciences, 2022, p. 02006.

[89] S. Raghuram, A. S. Bharadwaj, S. Deepika, M. S. Khadabadi, and A. Jayaprakash, "Digital implementation of the softmax activation function and the inverse softmax function," in *2022 4th international conference on circuits, control, communication and computing (I4C)*. IEEE, 2022, pp. 64–67.

[90] J. Liang, "Confusion matrix : Machine learning," *POGIL Activity Clearinghouse*, vol. 3, no. 4, 2022.

[91] B. B. Gupta and A. Dahiya, *Distributed Denial of Service (DDoS) Attacks : Classification, Attacks, Challenges and Countermeasures*. CRC Press, 2021.

[92] E. Fenil and P. M. Kumar, "Survey on ddos defense mechanisms," *Concurrency and Computation : Practice and Experience*, vol. 32, no. 4, p. e5114, 2020.

[93] R. Singh and T. P. Sharma, "Present status of distributed denial of service (ddos) attacks in internet world," *International Journal of Mathematical, Engineering and Management Sciences*, vol. 4, no. 4, p. 1008, 2019.

[94] M. Snehi and A. Bhandari, "Vulnerability retrospection of security solutions for software-defined cyber-physical system against ddos and iot-ddos attacks," *Computer Science Review*, vol. 40, p. 100371, 2021.

[95] M. D. Mauro, G. Galatro, G. Fortino *et al.*, "Supervised feature selection techniques in network intrusion detection : A critical review," *Engineering Applications of Artificial Intelligence*, vol. 101, p. 104216, 2021.

[96] F. Kamalov, S. Moussa, Z. E. Khatib *et al.*, "Orthogonal variance-based feature selection for intrusion detection systems," in *2021 International Symposium on Networks, Computers and Communications (ISNCC)*. IEEE, 2021, pp. 1–5.

[97] Y. Wei, J. Jang-Jaccard, F. Sabrina *et al.*, "Ae-mlp : A hybrid deep learning approach for ddos detection and classification," *IEEE Access*, vol. 9, pp. 146 810–146 821, 2021.

[98] V. Odumuyiwa and R. Alabi, "Ddos detection on internet of things using unsupervised algorithms," *Journal of Cyber Security and Mobility*, pp. 569–592, 2021.

[99] A. E. Cil, K. Yildiz, and A. Buldu, "Detection of ddos attacks with feed forward based deep neural network model," *Expert Systems with Applications*, vol. 169, p. 114520, 2021.

[100] G. C. Amaizu, C. I. Nwakanma, S. Bhardwaj *et al.*, "Composite and efficient ddos attack detection framework for b5g networks," *Computer Networks*, vol. 188, p. 107871, 2021.

[101] T. Khempetcg and P. Wuttidittachotti, "Ddos attack detection using deep learning," *IAES International Journal of Artificial Intelligence*, vol. 10, no. 2, p. 382, 2021.

[102] F. Hussain, S. G. Abbas, M. Husnain *et al.*, "Iot dos and ddos attack detection using resnet," in *2020 IEEE 23rd International Multitopic Conference (INMIC)*. IEEE, 2020, pp. 1–6.

[103] Y. Jia, F. Zhong, A. Alrawais *et al.*, "Flowguard : An intelligent edge defense mechanism against iot ddos attacks," *IEEE Internet of Things Journal*, vol. 7, no. 10, pp. 9552–9562, 2020.

[104] M. Shurman, R. M. Khrais, A. A. Yateem *et al.*, "Dos and ddos attack detection using deep learning and ids," *Int. Arab J. Inf. Technol.*, vol. 17, no. 4A, pp. 655–661, 2020.

[105] J. Li, M. Liu, Z. Xue *et al.*, "Rtvd : A real-time volumetric detection scheme for ddos in the internet of things," *IEEE Access*, vol. 8, pp. 36 191–36 201, 2020.

[106] I. Sharafaldin, A. H. Lashkari, S. Hakak *et al.*, "Developing realistic distributed denial of service (ddos) attack dataset and taxonomy," in *2019 International Carnahan Conference on Security Technology (ICCST)*. IEEE, 2019, pp. 1–8.

[107] D. Javeed, T. Gao, and M. T. Khan, "Sdn-enabled hybrid dl-driven framework for the detection of emerging cyber threats in iot," *Electronics*, vol. 10, no. 8, p. 918, 2021.

[108] H. A. Alamri and V. Thayananthan, "Bandwidth control mechanism and extreme gradient boosting algorithm for protecting software-defined networks against ddos attacks," *IEEE Access*, vol. 8, pp. 194 269–194 288, 2020.

[109] M. V. Assis, L. F. Carvalho, J. Lloret *et al.*, "A gru deep learning system against attacks in software defined networks," *Journal of Network and Computer Applications*, vol. 177, p. 102942, 2021.

[110] Z. Wu, Q. Xu, J. Wang *et al.*, "Low-rate ddos attack detection based on factorization machine in software defined network," *IEEE Access*, vol. 8, pp. 17 404–17 418, 2020.

[111] L. Yang and H. Zhao, "Ddos attack identification and defense using sdn based on machine learning method," in *2018 15th International Symposium on Pervasive Systems, Algorithms and Networks (I-SPAN)*. IEEE, 2018, pp. 174–178.

[112] K. M. Sudar and P. Deepalakshmi, "Flow-based detection and mitigation of low-rate ddos attack in sdn environment using machine learning techniques," in *IoT and Analytics for Sensor Networks : Proceedings of ICWSNUCA 2021*. Springer Singapore, 2022, pp. 193–205.

[113] A. A. Alashhab, M. S. M. Zahid, A. Muneer *et al.*, "Low-rate ddos attack detection using deep learning for sdn-enabled iot networks," *International Journal of Advanced Computer Science and Applications*, vol. 13, no. 11, 2022.

[114] D. Ravikumar, "Towards enhancement of machine learning techniques using cse-cic-ids2018 cybersecurity dataset," Ph.D. dissertation, Rochester Institute of Technology, 2021.

[115] M. A. Ferrag, L. Maglaras, S. Moscho *et al.*, "Deep learning for cyber security intrusion detection : Approaches, datasets, and comparative study," *Journal of Information Security and Applications*, vol. 50, p. 102419, 2020.

[116] B. B. Jia and M. L. Zhang, "Multi-dimensional classification via sparse label encoding," in *International Conference on Machine Learning*. PMLR, 2021, pp. 4917–4926.

[117] S. Rashka and V. Mirdzhalili, *Machine Learning and Deep Learning with Python, scikit-learn, and TensorFlow 2*. Birmingham, Mumbai : Packt, 2020.

[118] M. I. Tariq, N. A. Memon, S. Ahmed *et al.*, "A review of deep learning security and privacy defensive techniques," *Mobile Information Systems*, vol. 2020, pp. 1–18, 2020.

[119] B. Moons, D. Bankman, and M. Verhelst, *Embedded Deep Learning*. Springer, 2019.

[120] U. Michelucci, *Applied Deep Learning with TensorFlow 2 : Learn to Implement Advanced Deep Learning Techniques with Python*. Apress, 2022.

[121] S. Greengard, "Ai rewrites coding," *Communications of the ACM*, vol. 66, no. 4, pp. 12–14, 2023.

[122] S. Raschka, J. Patterson, and C. Nolet, "Machine learning in python : Main developments and technology trends in data science, machine learning, and artificial intelligence," *Information*, vol. 11, no. 4, p. 193, 2020.

[123] N. Silaparasetty, *Machine Learning Concepts with Python and the Jupyter Notebook Environment*. Berkeley, CA, USA : Apress, 2020.

[124] M. A. Ferrag, O. Friha, D. Hamouda *et al.*, "Edge-iiotset : A new comprehensive realistic cyber security dataset of iot and iiot applications for centralized and federated learning," *IEEE Access*, vol. 10, pp. 40 281–40 306, 2022.

[125] V. Borhade, A. Nayak, and R. Dakshayani, "Intrusion detection : A machine learning approach," in *Advanced Computing Technologies and Applications : Proceedings of 2nd International Conference on Advanced Computing Technologies and Applications (ICACTA 2020).* Springer, 2020, pp. 555–561.

[126] R. Sahani, Shatabdinalini, C. Rout, J. C. Badajena, A. Jena, and H. Das, "Classification of intrusion detection using data mining techniques," in *Progress in Computing, Analytics and Networking : Proceedings of ICCAN 2017.* Springer, 2018, pp. 753–764.

[127] M. Abdalla, X. Boyen, C. Chevalier, and D. Pointcheval, "Distributed public-key cryptography from weak secrets," in *Public Key Cryptography–PKC 2009 : 12th International Conference on Practice and Theory in Public Key Cryptography, Irvine, CA, USA, March 18-20, 2009. Proceedings 12.* Springer, 2009, pp. 139–159.

[128] U. Somani, K. Lakhani, and M. Mundra, "Implementing digital signature with rsa encryption algorithm to enhance the data security of cloud in cloud computing," in *2010 First International Conference on Parallel, Distributed and Grid Computing (PDGC 2010).* IEEE, 2010, pp. 211–216.

[129] D. Denning, "An intrusion-detection model," *IEEE Transactions on Software Engineering*, no. 2, pp. 222–232, 1987.

[130] M. Aydin, A. Zaim, and K. Ceylan, "A hybrid intrusion detection system design for computer network security," *Computers & Electrical Engineering*, vol. 35, no. 3, pp. 517–526, 2009.

[131] D. Anderson, T. Lunt, H. Javitz, A. Tamaru, and A. Valdes, "Detecting unusual program behavior using the nides statistical component," IDS Report SRI Project 2596, Tech. Rep., 1995.

[132] H.-J. Liao, C.-H. Lin, Y.-C. Lin, and K.-Y. Tung, "Intrusion detection system : A comprehensive review," *Journal of Network and Computer Applications*, vol. 36, no. 1, pp. 16–24, 2013.

[133] A. Sung and S. Mukkamala, "Identifying important features for intrusion detection using support vector machines and neural networks," in *2003 Symposium on Applications and the Internet.* IEEE, 2003, pp. 209–216.

[134] E. Hodo, X. Bellekens, A. Hamilton, C. Tachtatzis, and R. Atkinson, "Shallow and deep networks intrusion detection system : A taxonomy and survey," *arXiv preprint arXiv :1701.02145*, 2017.

[135] G. Karatas, O. Demir, and O. Sahingoz, "Deep learning in intrusion detection systems," in *2018 International Congress on Big Data, Deep Learning and Fighting Cyber Terrorism (IBIGDELFT).* IEEE, 2018, pp. 113–116.

[136] R. Farhan, A. Maolood, and N. Hassan, "Performance analysis of flow-based attacks detection on cse-cic-ids2018 dataset using deep learning," *Indonesian*

Journal of Electrical Engineering and Computer Science, vol. 20, no. 3, p. 1413, 2020.

[137] Y. Zhang, Y. Zhang, N. Zhang, and M. Xiao, "A network intrusion detection method based on deep learning with higher accuracy," *Procedia Computer Science*, vol. 174, pp. 50–54, 2020.

[138] S. Gamage and J. Samarabandu, "Deep learning methods in network intrusion detection : A survey and an objective comparison," *Journal of Network and Computer Applications*, vol. 169, p. 102767, 2020.

[139] V. Kanimozhi and T. Jacob, "Artificial intelligence out flanks all other machine learning classifiers in network intrusion detection system on the realistic cyber dataset cse-cic-ids2018 using cloud computing," *ICT Express*, vol. 7, no. 3, pp. 366–370, 2021.

[140] I. Sohn, "Deep belief network based intrusion detection techniques : A survey," *Expert Systems with Applications*, vol. 167, p. 114170, 2021.

[141] E. Abdallah, A. Otoom *et al.*, "Intrusion detection systems using supervised machine learning techniques : a survey," *Procedia Computer Science*, vol. 201, pp. 205–212, 2022.

[142] M. Sarhan, S. Layeghy, N. Moustafa, M. Gallagher, and M. Portmann, "Feature extraction for machine learning-based intrusion detection in iot networks," *Digital Communications and Networks*, 2022.

[143] B. Farhan and A. Jasim, "Performance analysis of intrusion detection for deep learning model based on cse-cic-ids2018 dataset," *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 26, no. 2, pp. 1165–1172, 2022.

[144] V. Hnamte and J. Hussain, "Dcnnbilstm : An efficient hybrid deep learning-based intrusion detection system," *Telematics and Informatics Reports*, vol. 10, p. 100053, 2023.

[145] R. Elsayed, R. Hamada, M. Abdalla, and S. Elsaid, "Securing iot and sdn systems using deep-learning based automatic intrusion detection," *Ain Shams Engineering Journal*, p. 102211, 2023.

[146] N. Saran and N. Kesswani, "A comparative study of supervised machine learning classifiers for intrusion detection in internet of things," *Procedia Computer Science*, vol. 218, pp. 2049–2057, 2023.

[147] S. Alzughaibi and S. E. Khediri, "A cloud intrusion detection systems based on dnn using backpropagation and pso on the cse-cic-ids2018 dataset," *Applied Sciences*, vol. 13, no. 4, p. 2276, 2023.

[148] M. Yadav and R. Kalpana, "Data preprocessing for intrusion detection system using encoding and normalization approaches," in *2019 11th International Conference on Advanced Computing (ICoAC)*. IEEE, 2019, pp. 265–269.

[149] J. Luengo, D. García-Gil, S. Ramirez-Gallego, S. Garcia, and F. Herrera, *Big Data Preprocessing*. Cham : Springer, 2020.

[150] G. Naidu, T. Zuva, and E. M. Sibanda, "A review of evaluation metrics in machine learning algorithms," in *Computer science on-line conference*. Springer, 2023, pp. 15–25.

[151] N. Manaswi, "Understanding and working with keras," in *Deep Learning with Applications Using Python : Chatbots and Face, Object, and Speech Recognition with TensorFlow and Keras*. Springer, 2018, pp. 31–43.

[152] Y. Bengio, A. Courville, and P. Vincent, "Representation learning : A review and new perspectives," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 8, pp. 1798–1828, 2013.

[153] P. Okenyi and T. Owens, "On the anatomy of human hacking," *Information Systems Security*, vol. 16, no. 6, pp. 302–314, 2007.

Résumé

Cette thèse traite de la sécurité des réseaux informatiques et de l'application du Deep Learning à la détection d'intrusions. Elle s'ouvre sur une présentation des fondements de la cybersécurité, des principales menaces comme les attaques DoS, DDoS, brute force, injections SQL et botnets, ainsi que des mécanismes de protection tels que les pare-feux, le cryptage, les VPN et les systèmes de détection d'intrusions (IDS). Elle introduit ensuite les concepts d'intelligence artificielle, de machine learning et de deep learning, en décrivant les architectures neuronales (ANN, CNN, RNN) et les métriques d'évaluation utilisées pour l'analyse et la classification des attaques. L'étude expérimentale porte sur la détection d'attaques DDoS à l'aide de modèles de deep learning appliqués aux datasets CSE-CIC-IDS2018 et Edge-IIoTSet, avec une comparaison des performances selon différents types de classification. Enfin, une approche améliorée de détection d'anomalies IDS est proposée, intégrant des techniques d'optimisation et de réduction de dimension afin d'accroître la précision et la robustesse du modèle. Les résultats démontrent l'efficacité du deep learning dans la détection automatisée des intrusions et ouvrent la voie à des modèles plus légers et adaptatifs pour les environnements IoT et Edge Computing.

Mots clés : Sécurité réseaux, Deep Learning, DDoS, Détection d'intrusion, Intelligence Artificielle, détection d'attaques, classification, datasets CSE-CIC-IDS20218 et Edge-IIoTSet.

Abstract

This thesis addresses the security of computer networks and the application of Deep Learning to intrusion detection. It begins with a presentation of the foundations of cybersecurity, the main threats such as DoS, DDoS, brute force attacks, SQL injections, and botnets, as well as protection mechanisms including firewalls, encryption, VPNs, and intrusion detection systems (IDS). It then introduces the concepts of artificial intelligence, machine learning, and deep learning, describing neural architectures (ANN, CNN, RNN) and evaluation metrics used for the analysis and classification of attacks. The experimental study focuses on detecting DDoS attacks using deep learning models applied to the CSE-CIC-IDS2018 and Edge-IIoTSet datasets, with a performance comparison across different types of classification. Finally, an enhanced IDS anomaly detection approach is proposed, integrating optimization and dimensionality reduction techniques to improve the models accuracy and robustness. The results demonstrate the effectiveness of deep learning in automated intrusion detection and pave the way for lighter and more adaptive models suitable for IoT and Edge Computing environments.

Keywords: Network security, deep learning, DDoS, intrusion detection, artificial intelligence, attack detection, classification, CSE-CIC-IDS20218 and Edge-IIoTSet datasets.

ملخص

تتناول هذه الأطروحة أمن الشبكات الحاسوبية وتطبيقات التعلم العميق في كشف التسلل. تُعد الهجمات من أخطر تهديدات الأمن السيبراني، ومنها ما (DoS) يُنفَّذ لأداء التخريبات، مثل هجمات الحرمان من الخدمة (DDoS) وهجمات الحرمان من الخدمة الموزعة وهجمات القوة العمياء، وضخ (IDS) تُعدّ أنظمة كشف التسلل .VPNأداة أولية للحماية، لكنّها محدودة في التعرف على الـ (firewalls) تعدّ جدران الحماية .SQL من أهم تقنيات (RNNs، CNNs، الدفاع الذكية لاكتشاف الأنماط الجديدة، وتتنوع بين شبكية ومضيفة، كما تختلف في خوارزمياتها. تسهم شبكات التعلم العميق مثل في تطور أنظمة الذكاء الاصطناعي، وتُستخدم في تحليل بيانات الهجمات. تعتمد هذه الأطروحة على مبدأ تعلم الآلة والتعلم العميق للكشف (ANNs تهدف الأطروحة إلى تقديم نموذج لكشف .Edge-IIoTSetو ،CSE-CIC-IDS2018 عن الهجمات خلال بيئات محاكاة باستخدام: مركز بيانات تُظهر النتائج .Edge-IIoTSet وأيضًا نموذج آخر يعتمد على ،CSE-CIC-IDS2018 اعتمادًا على مجموعة بيانات (DDoS) الهجمات العميقة وتقليل معدلات الخطأ IDS دقة مختلفة، مع غلبة النمذجة باستخدام الشبكات العميقة. يمثل العمل المقترح خطوة إضافية في تطوير أداء أنظمة .الخاطئ، ويسهم في حماية الأنظمة، وخاصة مستقبلًا مع تطور تقنيات إنترنت الأشياء والذكاء الاصطناعي

أنظمة كشف التسلل، الذكاء ،(DDoS) **الكلمات المفتاحية:** أمن الشبكات، التعلم العميق، شبكات الحاسوب، هجمات الحرمان من الخدمة الموزعة .IDS2018 و Edge-IIoTSet ،CSE-CIC الاصطناعي، تعلم الآلة، الشبكات العصبية، مجموعة بيانات